Video compression based on smart decoder

Dang Khoa Vo Nguyen

UNIVERSITE NICE-SOPHIA ANTIPOLIS

ECOLE DOCTORALE STIC

SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

T H E S E

pour l'obtention du grade de

Docteur en Sciences

de l'Université Nice-Sophia Antipolis

Mention : Automatique, Traitement du Signal et des Images

présentée et soutenue par

Dang Khoa VO NGUYEN

Compression vidéo basée sur l'exploitation d'un décodeur intelligent

Thèse dirigée par
Marc ANTONINI, Directeur de Recherche, CNRS
Equipe d'accueil : Laboratoire I3S, UNS-CNRS
et
Joël JUNG, Ingénieur de Recherche, Orange Labs
Equipe d'accueil : CVA, Orange Labs

soutenue le *18/12/2015*

Jury :

Amel BENAZZA-BENYAHIA	Ecole Supérieure des Communications de Tunis, Tunisie	Rapporteur
Marco CAGNAZZO	CNRS-Télécom ParisTech, France	Rapporteur
Marc ANTONINI	CNRS-Université Nice-Sophia Antipolis, France	Directeur de thèse
Joël JUNG	Orange Labs, France	Encadrant de thèse
Jean-Luc DUGELAY	Eurecom, France	Examinateur
Luce MORIN	INSA Rennes, France	Examinateur

To my beloved parents. . .

Remerciements

Tout d'abord, je tiens à exprimer ma sincère gratitude à mon encadrant de thèse, Dr. Joël JUNG, pour son soutien dans ma recherche pendant les trois années de doctorat, pour sa patience, sa motivation et son enthousiasme. C'est grâce à sa direction et son immense compétence que j'ai pu progresser dans ma recherche et dans la rédaction de ce manuscrit. Son oeil critique m'a été très précieux pour structurer le travail et pour améliorer la qualité des travaux réalisés.

Je tiens aussi à remercier mon directeur de thèse, Marc ANTONINI, pour son soutien, ses observations judicieuses, ses idées intéressantes et ses suggestions constructives. Qu'il trouve ici l'expression de ma profonde reconnaissance

Je tiens également à remercier Amel BENAZZA-BENYAHIA, professeur à l'Ecole Supérieure des Communications de Tunis, et Marco CAGNAZZO, maitre de conférence à CNRS-Télécom ParisTech, d'avoir accepté d'être les rapporteurs de ce manuscrit. Je remercie Jean-Luc DUGELAY, professeur à Eurecom, et Luce MORIN, maitre de conférence à INSA Rennes, d'avoir accepté d'évaluer ce travail de thèse.

Mes sincères remerciements vont aussi à toute l'équipe de Compression Vidéo Avancée d'Orange Labs. J'ai passé de très agréables moments avec vous lors des réunions d'équipes à Rennes. Je souhaite remercier particulièrement mon manager Didier GAUBIL pour ses encouragements ainsi que son aide pour des problèmes administratifs.

Je remercie mes collègues de bureau: Kartik, Elie, et Antoine pour les discussions stimulantes et pour tout le plaisir que nous avons eu au cours de ma thèse. J'ai beaucoup apprécié leur compétences et leur aide qui m'ont permis à progresser dans ma recherche.

Finalement, je tiens à remercier ma famille et mes parents pour m'avoir soutenir spirituellement tout au long de ma vie. Aucun mot ne peut exprimer ma gratitude envers ma mère qui a sacrifié toute sa vie pour ma réussite.

Table of Contents

List of Figures

List of Tables

List of Acronyms

3D-HEVC	Three-dimensional High Efficiency Video Coding
AMVP	Advanced Motion Vector Prediction
ARP	Advanced Residual Prediction
AVC	Advanced Video Coding
BD	Bjøntegaard Delta
CABAC	Context-Adaptive Binary Arithmetic Coding
CTC	Common Test Conditions
CTU	Coding Tree Unit
CU	Coding Unit
DCP	Disparity-Compensated Prediction
DMM	Depth Modeling Modes
DMVF	Dense Motion Vector Field
DV	Disparity Vector
GOP	Group Of Pictures
HEVC	High Efficiency Video Coding
HM	HEVC test Model (reference software for HEVC)
IVMP	Inter-View Motion Prediction
JCT	Joint Collaborative Team
MCP	Motion-Compensated Prediction
MPM	Most Probable Mode
MV	Motion Vector
OF	Optical Flow
PSNR	Peak Signal to Noise Ratio
PU	Prediction Unit
QP	Quantization Parameter
RD	Rate-Distortion
SDec	Smart Decoder
SP-IVMP	Sub-PU Inter-View Motion Prediction
TU	Transform Unit
WOF	Warped Optical Flow

Introduction

Context

During these last recent years, advances in technology make a revolution in the field of digital video and transform the world of multimedia by multiplying the available content, creating new methods for capturing, transmitting and viewing videos. Display resolution continue to be increased with the widespread adoption of larger resolutions like High-Definition (HD) and Ultra-High-Definition (UHD). Numerous television channels are currently offering higher resolution than the standard SD resolution of 720×576. The well-known DVD format is currently replaced by Blu-ray, a format that allows much larger video resolution by significantly increasing the storage capacity.

Furthermore, the rapid growth of video sharing platforms, such as YouTube which generates billion of video views per day, and the fast development of video on demand (VOD) services and Internet Protocol television (IPTV) networks are important factors contributing to the significant increase in video traffic within the Internet network. The forecast shows an even higher growth in video traffic in the near future. Indeed, according to recent studies conducted by Cisco (Cisco 2015), the global IP traffic in 2014 has increased more than fivefold in the past five years, and will increase nearly threefold over the next five years by 2019. The Internet video traffic will be 80 to 90 percent of all consumer Internet traffic in 2019. This ratio will be pushed further with the apparition of 3D super multi-view (3D SMV) video. Indeed, demand for 3D video is constantly increased and becomes very popular with the advertising of 3D cinema and devices dedicated for virtual reality such as 3D television or head-mounted display (HMD). Enhancing technologies on 2D video, such as high frame rate (HFR), high dynamic rate (HDR) or wide color gamut (HCG), also contribute a major part to the increase in the overall Internet video traffic.

Given the impact of video contents on the global Internet traffic, video compression becomes a very important challenge. Since digital video is one of the most demanding applications both in terms of space required for storage and bandwidth for distribution, a video requires to be compressed before being transmitted. Compression is the process by which large files are being reduced in size by deleting from these files all redundant information. Compression will therefore include determining these redundancies and eliminating them while preserving the capacity to reproduce the original files, intact or with minimal degradation.

It is in this context that a telecommunications company such as Orange has an interest in the development and the standardization of compression techniques in order to ensure that the best possible codec will be standardized, so that multimedia services become more accessible and continue to grow rapidly. This thesis has the objective to improve the coding efficiency of existing codecs by proposing methods that break away from conventional technologies. Entire work in this thesis was performed within the Advanced Video Coding (CVA) team of Orange Labs, in collaboration with the I3S laboratory of the university of Nice Sophia-Antipolis and the French National Centre for Scientific Research (CNRS).

Contributions

High Efficiency Video Coding (HEVC) recently becomes the video compression standard to succeed H.264/AVC. Considering HEVC as video coding basis, this thesis proposes to study a novel concept called Smart Decoder (SDec). A major part of our research is devoted to elaborating the SDec coding scheme and several of its practical applications. Several contributions are identified as follows:

– General encoding and decoding scheme based on the SDec technique are created to remove the limit of conventional coding schemes, which is related to the increased number of available coding modes. This approach is proposed with the objective to reduce the signaling cost of coding modes and associated coding parameters. It exploits a decoder that is able to simulate the encoder and to conduct the rate-distortion (R-D) competition in order to retrieve coding modes that are used by the encoder. In the SDec scheme, the coding mode of a block is not signaled in the bit stream, saving thus its signaling overhead.

– A first practical application of the SDec scheme is proposed, exploiting the Intra coding mode. Up to 35 Intra directions are put in the competition with each other during the SDec process. The Intra direction does not need to be signaled for a block encoded in SDec mode, reducing in consequence the signaling overhead.

– A second practical application of the SDec scheme where Intra 1D coding mode is exploited is proposed. Intra 1D mode has the particularity of having numerous parameters, making it hard to apply in real applications due to the costly signaling overhead. Integrating Intra 1D in the SDec scheme has the advantange of removing the signaling cost of its intrinsic parameters.

– A third practical application of the SDec scheme is proposed, where motion vector (MV) is re-estimated based on the causal information using a more relevant criterion. This re-estimated motion is then inherited to encode a block without being signaled in the bit stream. Two types of motion re-estimation are proposed, which are based respectively on block matching and optical flow techniques.

– A fourth practical application of the SDec scheme is proposed, where the combination of Intra, Intra 1D and Inter with re-estimated motion is used for the SDec process.

Another important contribution of the thesis is related to the long-term approach which continues to exploit a decoder able to perform complex processes in order to reduce the signaling of coding parameters. Our research for the long-term approach that uses machine learning in video coding is initiated by a work which aims to improve the Intra Most Probable Mode (MPM) signaling scheme, where it is observed that machine learning technique can be exploited to improve the video compression performance:

– An improved MPM signaling scheme that exploits a fourth value is proposed. Added MPM value is shown to improve the efficiency of the MPM scheme, increasing therefore the coding performance as the optimal Intra direction of a block is better predicted.

– Novel encoding and decoding schemes which exploit machine learning techniques to predict the optimal coding mode are proposed. Bit rate saving is obtained from correct predictions made by the block classifier.

All these proposed methods are integrated within the state of the art HEVC codec, the latest standard for video coding.

Structure of the manuscript

This manuscript is organized into two main parts following an introduction to 2D and 3D video coding. The first part presents the concept SDec and its different practical applications. The second part describes the innovative long-term approach which makes use of machine learning techniques to reduce the signaling of coding parameters. We list in more detail all the chapters of the manuscript as follows:

– Chapter 1 presents an overview of both 2D and 3D video standards, detailing their general structure and coding tools that are mentioned in our works. The state of the art of several techniques related to our research, such as Intra 1D or machine learning based video coding techniques, are also given.

Part 1: SDec scheme and its practical applications

– Chapter 2 introduces the concept of "Smart Decoder" (SDec). In this chapter, the general outline of the proposed SDec scheme, including its encoding and decoding processes, is given. Advantages and drawbacks of the SDec scheme are then described. Particular characteristics of this coding scheme are also discussed.

– Chapter 3, 4, 5 and 6 present four practical applications of the SDec coding scheme, respectively using Intra, Intra 1D, Inter with re-estimated motion parameters, and a combination of all those three coding modes. They are implemented in the HEVC test model software (HM). In each chapter, the description and experimental results of the proposed application are given, following several studies that refer to different intrinsic parameters associated to the SDec scheme.

Part 2: Machine learning based video coding

- Chapter 7 describes the proposed improvement of the Intra MPM signaling scheme. Experimental results are presented, followed by a perspective that introduces the use of machine learning in video coding.
- Chapter 8 introduces the proposed video coding scheme that exploits machine learning techniques to reduce the signaling overhead. The background of our work is first provided, describing the generalities of conventional coding schemes and machine learning based coding schemes that exist in the state of the art. The proposed coding method is then described. Practical applications with experimental results are finally given.

A summary of the proposed methods and their associated results are given at the end of this manuscript as conclusion, followed by some perspectives for future work.

1 Video coding and HEVC standard

In the first chapter, an introduction to video coding is presented. Our works are based on High Efficiency Video Coding (HEVC), the new standard that replaces H.264/AVC, offering an average reduction of 50% in bit rate with the same subjective image quality over its predecessor. Its 3D counterpart, the 3D-HEVC standard, is also introduced. An overview of both standards is presented, addressing their general structure and principal tools. Introduction to coding techniques exploiting a complex decoder is then given. The state of the art on improvement techniques related to our works, such as Intra 1D or machine learning based video coding techniques, will be described as well. This chapter is considered as a basis for our further works which aim to improve the existing video coding scheme.

1.1 Context

Since 2003, H.264/AVC (ITU-T 2003) is one of the most commonly used formats for recording, compression, and distribution of video content. Compared with H.262/MPEG-2 (ITU-T and ISO/IEC-JTC1 1994), which is one of its predecessors and standardized in 1994, H.264/AVC allows a significant compression performance of 50% in bit rate reduction for a similar encoded video quality. About ten years later, there is a need to come up with a new video coding standard that aims to outperform H.264/AVC.

In 2013, HEVC, the successor to H.264/AVC, is released as the new video compression standard, which was jointly developed by ISO/IEC MPEG and ITU-T VCEG. Again, the performance requirement is achieved. With a compression ratio of 50% higher than the H.264/AVC at the same visual quality, HEVC is expected to revolutionize several fields, for example:

- allowing the deployment of new video formats (ultra-HD, 4K, 8K, 3D...),
- widening the eligibility for IPTV (SD and HD formats).

1.2 HEVC standard

Similar to previous video coding standards, HEVC is based on a hybrid approach exploiting the spatial (Intra) and temporal (Inter) redundancies of the video signal using multiple choice coding competition (Sullivan et al. 2012). The block diagram of a typical HEVC video encoder is described in figure 1.1. We detail in following sections some technical features and characteristics of the HEVC standard. First, the overall coding structure is presented. Then, important tools related to our works are described, including Intra, Inter predictions and Context-Adaptive Binary Arithmetic Coding (CABAC).

1.2.1 Coding stucture

HEVC uses a Group of Pictures (GOP) structure to group several successive frames within a video sequence where Intra and Inter frames are arranged in a specific order. Thus a coded video stream consists of successive GOPs. In HEVC test model software (HM), each GOP can contain three following types of frames:

- **I-frame**: Intra frame that is encoded independently of all other frames.
- **P-frame**: Inter predictive frame that contains motion compensated difference information relative to previously decoded frames. A P-frame can only refer to one other frame as reference, found in the reference picture list L0.
- **B-frame**: Inter bipredictive frame that contains motion compensated difference information relative to previously decoded frames. Unlike P-frame, it can refer to two other frames, found respectively in the reference picture lists L0 and L1.

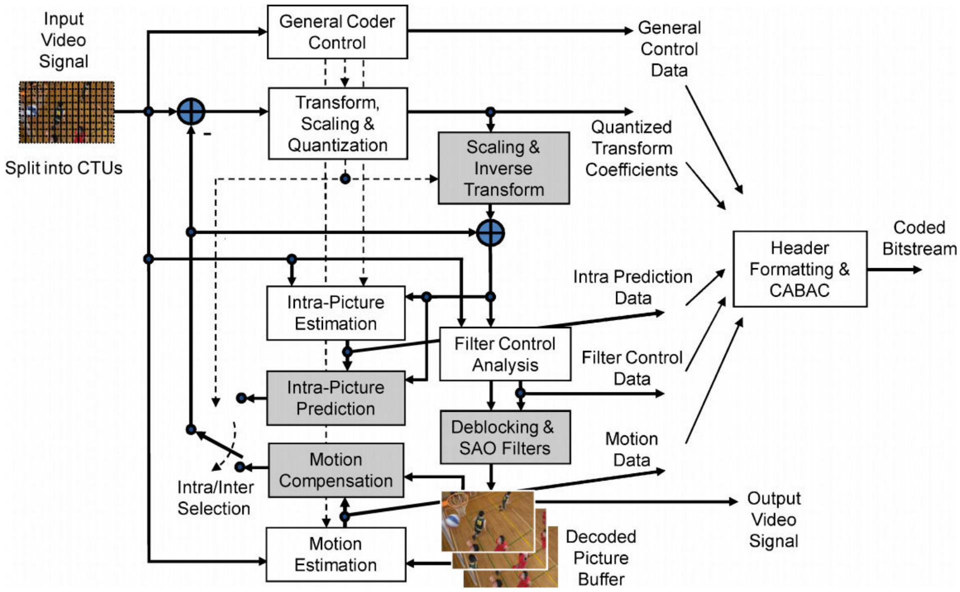


Figure 1.1 – *Typical HEVC video encoder (with decoder modeling elements shaded in light gray) (Source: (Sullivan et al. 2012)).*

During the development of HEVC, three GOP structures are used for purpose of testing. They are depicted in figure 1.2.

- **All Intra** (AI): all frames are encoded as I-frames. It is typically used for low complexity encoders that can afford large bandwidth (e.g. digital still camera, video surveillance).
- **Low delay** (LD): only the first frame is encoded as I-frame. Other frames are either P or B-frames, which are in their normal order of processing in the bit stream. Thus a frame can only have previously decoded frames as references. It is typically used for videoconferencing.
- **Random access** (RA): an I-frame is inserted every one seconde. Frame pictures are not in their normal order of processing in the bit stream. Thus past and future reference frames are allowed. This GOP structure is often used when a small delay due to frames buffering is not an issue.

Within each frame, the encoding process is conducted on a block basis, with three following basic processing units that form the core of the coding layer:

- **Coding unit** (CU): defines the basic unit for the whole encoding process. The quad-tree syntax allows to split a CU in smaller CUs.
- **Prediction unit** (PU): defines the basic unit for the prediction. A PU can have different partitionings as follows: $2N\times2N$, $2N\times N$, $N\times2N$, $N\times N$, $2N\times nU$, $2N\times nD$, $nL\times2N$ and $nR\times2N$. PUs are contained in CUs.
- **Transform unit** (TU): defines the basic unit on which transform and quantization are applied. The partitioning of TUs is independent from PUs but is still contained in a CU.

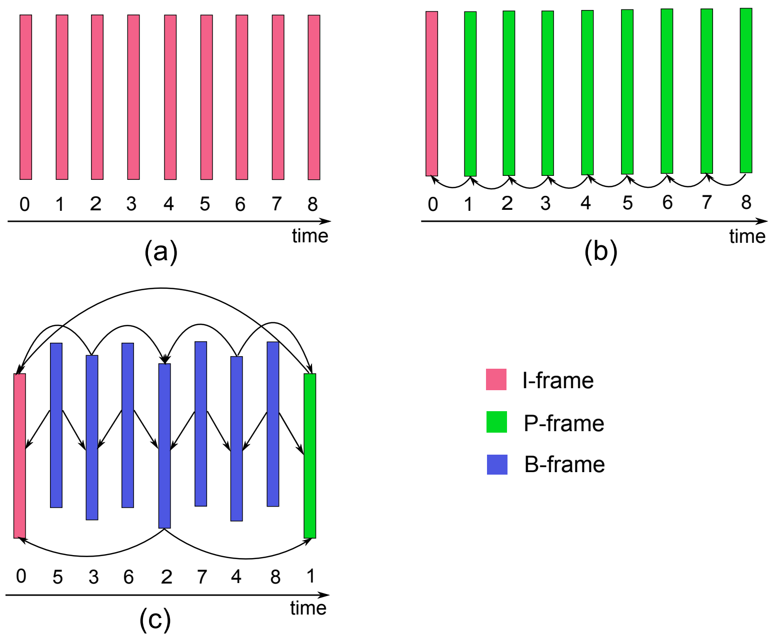


Figure 1.2 – *GOP structures with different types of frame highlighted in colors. Arrows indicate references of a frame. The number under each frame indicates its order of processing. (a) AI. (b) LP. (c) RA.*

Practically, using HM software, the size of the *largest coding unit*, also known as *Coding Tree Block* (CTB) and the maximum depth of partitioning for CUs and TUs are defined in a configuration file as encoding parameters. The size of each specific CU, PU or TU is then recursively determined during the encoding, based on a choice criterion that maximizes the coding efficiency. An example of partitioning a 64×64 CTB is given in figure 1.3.

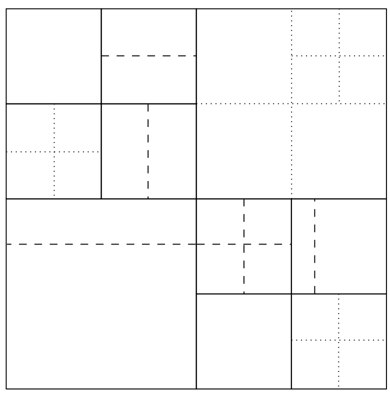


Figure 1.3 – *Subdivision of a CTB into CUs, PUs and TUs. Solid lines indicates CU boundaries. Dashed lines indicates PU boundaries. Dotted lines indicates TU boundaries.*

1.2.2 Intra prediction

Intra prediction in HEVC uses a competition of up to 35 predictors for luminance component: DC, Planar and 33 directions as displayed in the left of figure 1.4. All predictors give a prediction of the current PU using reference pixels located directly in the left and above neighboring regions.

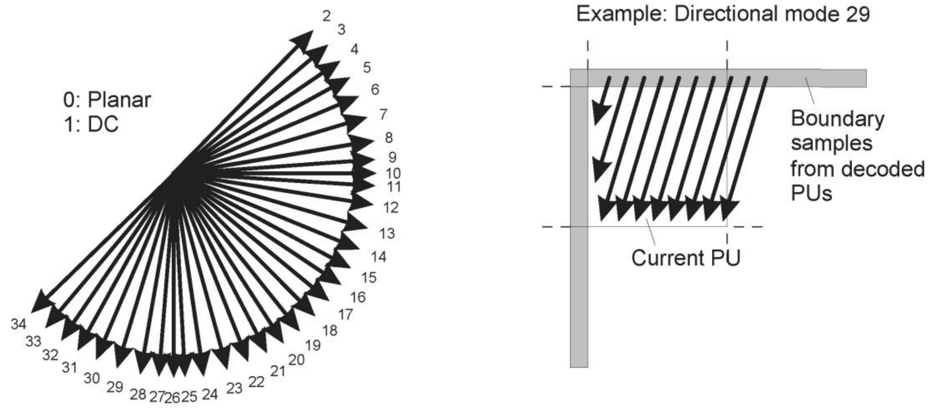


Figure 1.4 – *Modes and directional orientations for Intra prediction (Source: (Sullivan et al. 2012)).*

The Intra DC prediction uses an average value of reference samples for the prediction of the whole PU, which is best suited for uniform regions, while Intra Planar uses average values of two linear interpolations based on four corner reference samples to prevent discontinuities along the PU boundaries.

For directional predictors, reference samples are projected along a given directionality to construct the predicted PU, as shown in the right of figure 1.4. For a PU of size $N \times N$, a total of up to $4N+1$ spatially neighboring samples may be used for the prediction, which consist of N pixels for each of Left, Bottom Left, Above and Above Right sides, and 1 pixel in the Above Left corner. The 33 angles are designed in a way that provides denser coverage for near-horizontal and near-vertical angles to reflect the observed statistical prevalence of the angles and the effectiveness of the signal prediction processing.

For chrominance components, the number of Intra predictors is restrained compared to luminance. Only five predictors are used: horizontal, vertical, Planar, DC and the luma prediction mode.

An Intra predicted CU has two possible PU partition sizes: $2N \times 2N$ and $N \times N$. The partition size $N \times N$ is only used when current CU size is equal to the minimum CU size.

Signaling the Intra prediction mode for luminance consists of encoding the Intra predictor index selected among 35 values. The Intra Most Probable Mode (MPM) is introduced in HEVC to efficiently signal the predictor by making use of values that are most likely to be selected. Three MPM values are elaborated based on the already encoded Left and Above PUs.

The list of Intra MPM candidates is constructed as follows: let A, B be the Intra predictor of the Left and Above PUs. If they are not available or not encoded in Intra mode, A and B are set to the DC predictor. Let *CandList* be the list of three MPM values to be computed. There are in total five possible cases for the construction of *CandList* based on A and B as follows:

- If $A = B$
  - If (A angular), CandList = $\{A, A-1, A+1\}$ (case 1)

  - ◦ Else, $CandList$ = {Planar, DC, vertical} (case 2)
- – If $A \neq B$,
  - ◦ If ($A \neq Planar$ and $B \neq Planar$), $CandList$ = {A, B, Planar} (case 3)
  - ◦ Else if ($A \neq DC$ and $B \neq DC$), CandList = {A, B, DC} (case 4)
  - ◦ Else, $CandList$ = {A, B, vertical} (case 5)

If the Intra predictor to be encoded matches one of the MPM candidates, a 1-bit syntax element called $prev\_intra\_luma\_pred\_flag$ is signaled to indicate the use of the Intra MPM technique, followed by the signaling of the syntax element mpm_idx indicating the index of selected MPM candidate, which can be encoded by 2 bits at most. In the case where no MPM candidate matches the Intra predictor to be encoded, the latter will be signaled in the bit stream by the syntax element $rem\_intra\_lum\_pred\_mode$ using 5 bits. The signaling scheme concerning the Intra MPM technique is summarized in figure 1.5.

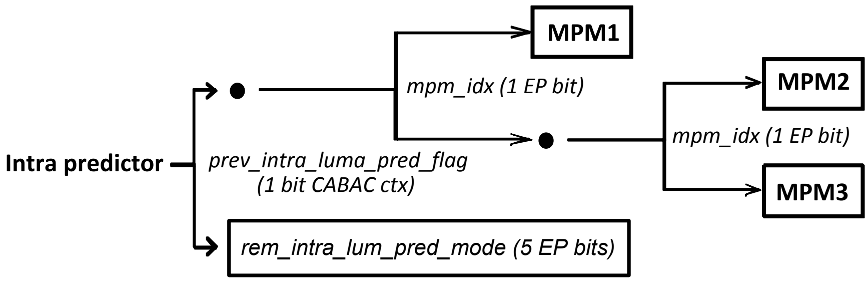


Figure 1.5 – *Intra MPM signaling scheme.*

1.2.3 Inter prediction

The Inter prediction in HEVC includes two main components: Motion Compensation (MC) and the encoding of motion parameters. The Motion Estimation (ME), the process of determining the *motion vector* (MV) pointing to a block in an already decoded frame that best describes a PU to be encoded in the current frame, is not normative. The MV computed by the ME process is then used in the MC process which calculates the motion compensated difference between the current PU and the reference PU pointed by the MV. Note that there are particular temporal coding modes that do not need the ME process to acquire MVs, such as Skip and Merge modes which inherit MVs from specific causal blocks.

An illustrated example is shown in figure 1.6. On the left, the ME technique determines the optimal MV that points to a zone P', located in a reference frame at instant t_{i-1} using block matching process. P' is computed by minimizing both the distortion to current PU P located in current frame at instant t_i and the cost to signal the MV (Rate-Distortion criterion). Then, on the right, the MC is called to calculate the motion compensated difference between P and P' that will be signaled in the bit stream.

The precision of the ME and MC processes in HEVC are at the level of precision of 1/4 pixel for luminance, thanks to the use of a 8-tap and 7-tap interpolation filters respectively for half-pel and quarter-pel positions. A precision of 1/8 pixel for chrominance is obtained using a 4-tap filter for

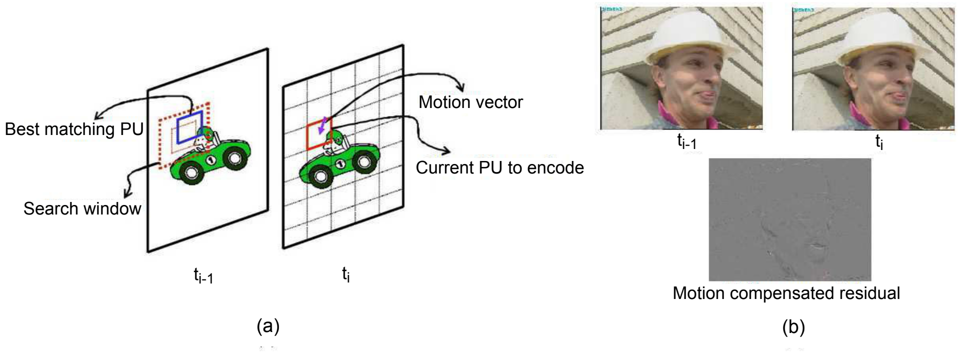


Figure 1.6 – *(a) Motion estimation. (b) Motion compensation.*

eighth-pel positions.

The encoding of motion parameters is supported by two principal tools in HEVC: the *Advanced Motion Vector Prediction* (AMVP) which encodes the difference between the selected MV and its predictor, and the *Merge* mode (Sullivan et al. 2012, Helle et al. 2012) which only encodes the index of inherited MV among a list of predictors. Both tools consist of techniques to select optimal MV while reducing its signaling overhead.

In AMVP, a list of two candidates for the MV predictor (MVP) is first constructed. Both candidates are selected among the MVs of the PUs covering different positions as shown in figure 1.7.

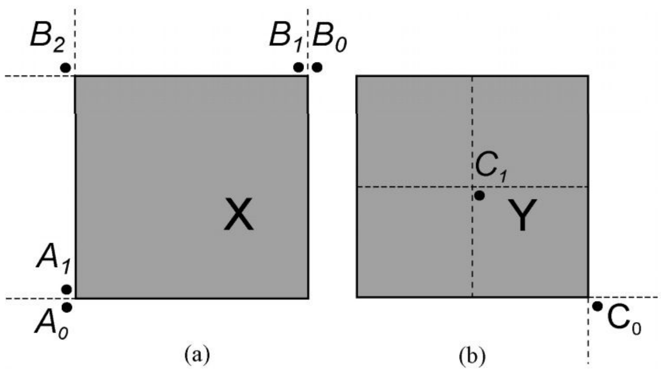


Figure 1.7 – *Positions of possible MVP candidates for a current PU X. (a) Spatial MVP candidate positions. (b) Temporal MVP candidate positions, where Y is the collocated block of X in a reference picture (Source: (Helle et al. 2012)).*

By verifying the PU at position A0 and A1, the first MV which is found will be inserted in the MVP candidates list as the first candidate. In the same manner, the second candidate is derived from B0, B1 and B2. If the list is not yet full or both the candidates found are identical, a temporal candidate derived from C0 and C1 will be added to the list. Zero MVs are used to fill the list in the end if it still contains less than two candidates.

Using the MVP candidates list, instead of sending the MV in the bit stream which is very costly, only the residual between the MV and the MVP is a along with an index signaling which MVP candidate out of the two was selected. This reduces indeed the amount of bits dedicated for signaling motion parameter.

The second important tool for Inter prediction is the Merge mode which derives the motion information, including motion vectors and indices of reference frames, from spatially or temporally neighboring blocks. A set of five possible MV candidates is thus constructed.

More precisely, the Merge MV candidates set consists of spatial neighbor candidates, a temporal candidate, and combined bi-predictive candidates. Similar as in AMVP, spatial candidates are found by checking available motion information of PUs covering in the order the positions A1, B1, B0, A0, B2. After validating the spatial candidates, a pruning process is performed to remove any redundancy. A maximum of four spatial candidates is allowed. A temporal candidate is then added by checking positions C0, C1. If the set contains less than five candidates, additional combined bi-predictive candidates are generated.

Using this Merge candidates set, the motion information, which is inherited from an already decoded block, can be simply transmitted by an index indicating its position in the set, saving its signaling cost.

We also remark that, unlike Intra predicted CUs, Inter predicted CUs can have all possible PU partitioning modes, in particular asymmetric ones. This flexibility in PU partitioning allows irregular image patterns to be efficiently represented without requiring further CU splitting.

1.2.4 Residual coding

After that the current PU is predicted with Intra or Inter prediction, its residual which corresponds to the difference between the original and predicted current PU is computed. The residual block will be subjected to a transformation in order to eliminate the spatial redundancies, followed by a quantization which is a lossy process that impacts on the quality of encoded video. The Quantization Parameter (QP) is used to control the quantization process. It has values ranged from 0 (highest quality) to 51 (lowest quality).

The block is then scanned to collect the transformed and quantized residual coefficients. The rearranged coefficients form therefore a 1D list of coefficients that will be efficiently compressed using lossless coding techniques and finally signaled in the bit stream.

We remark that in case where there is no texture residual for the CU being encoded and if Merge mode is selected to inherit MV, it is equivalent to Skip mode. In this specific case, only a Skip flag and the corresponding Merge index are transmitted.

1.2.5 Other tools

The residual coefficients and all syntax elements are transmitted in the bit stream using an entropy coding method called *Context-Adaptive Binary Arithmetic Coding* (CABAC) (Marpe et al. 2003). It is based on arithmetic coding, with following changes to adapt to the needs of video coding standards:

– It encodes binary symbols, which keeps the complexity low and allows probability modelling for more frequently used bits of any symbol,

– The probability models are selected adaptively based on local context, allowing better modelling of probabilities, because coding modes are usually locally well correlated,
– It uses a multiplication-free range division by the use of quantized probability ranges and probability states.

CABAC has multiple probability modes for different contexts. It first converts all non-binary symbols to binary. Then, for each bit, the coder selects which probability model to use, then uses information from nearby elements to optimize the probability estimate. Arithmetic coding is finally applied to compress the data.

In HEVC, encoding a syntax element with CABAC can take one among two forms as follows:
– **Without context** (bypass mode): coded bits are encoded using bypass coding engine without modeling the context. It is therefore very fast and parallelizable. However, compression level is often less performant than using context.
– **With context**: bits are encoded according to a selected probability model which is updated based on actual values coded in the bit stream. The whole process is thus slower than when using bypass mode. Before encoding each frame, contexts probability are reset to predefined initial values which are often different according to whether it is an I-, P- or B-frame. Those initial values are linear regarding the QP used for the encoding process, containing two parameters *slope* and *offset*, each coded with 8 bits (0-255). For an example of how an initial value of a context concerning a flag element is computed for CABAC encoding, figure 1.8 represents the probability of the flag element being equal to 1 according to the QP values when encoded in different coding qualities. Linear regression is used so that the slope and the offset of the line can be finally computed.

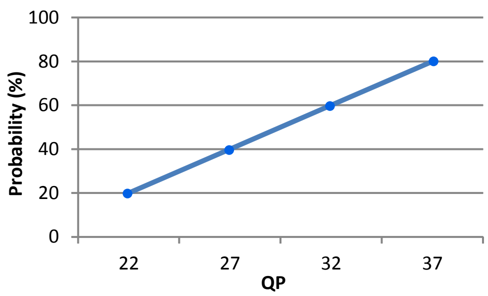


Figure 1.8 – *Computing the context initial value for a flag element involves plotting the probability of this flag being equal to 1.*

Appropriate selection of CABAC context is known to be a key factor to improve the efficiency of CABAC coding. Practically, the depth of the coding tree or data of spatially neighboring blocks is exploited to derive the context models of various syntax elements. Dependencies between coded data are also carefully considered to enable further throughput maximization.

Beside CABAC, it is also interesting to note that in HEVC, there are two processing steps, a *Deblocking Filter* (DBF) followed by a *Sample Adaptive Offset* (SAO) filter, that are applied to the reconstructed samples before writing them into the decoded picture buffer. The DBF is intended to reduce the blocking artifacts due to block-based coding, while the SAO modifies the decoded samples by conditionally adding an offset value to each sample, based on values in look-up tables transmitted by the encoder.

1.3 3D-HEVC standard

In this section, we introduce an overview of 3D-HEVC, the standard for coding 3D video. This is the 3D counterpart of HEVC which is for 2D video coding. We will also present some important tools specific for 3D coding purpose, such as Disparity-Compensated Prediction, Inter-View Motion Prediction and Advanced Residual Prediction. The understanding of these coding tools is required since they will be mentioned in our 3D related works.

1.3.1 Coding structure

3D-HEVC is designed to encode depth-based 3D video which consists of several texture views associated with their depth views. Among those texture views, there is a *base view* that is encoded independently from the others which are called *dependent views*. For each view, its depth map is a grayscale luminance only image which maps each pixel in the associated texture frame to a certain distance from the camera as depicted in figure 1.9.

Figure 1.9 – *A depth map with its associated texture frame. The higher the luminance value (brighter), the closer the object is to the camera.*

Depth maps are not displayed on screen but are used to synthesize intermediate views between two main views by interpolation technique. Figure 1.10 shows three intermediate views ($V2, V3, V4$) between two main views ($V1, V5$) synthesized using their texture and depth.

Since our research refers only to the coding of video texture, tools dedicated to encode depth map are not further detailed.

In 3D-HEVC, the base view is coded first, followed by dependent views. The base view is coded with unmodified HEVC codec. For de-

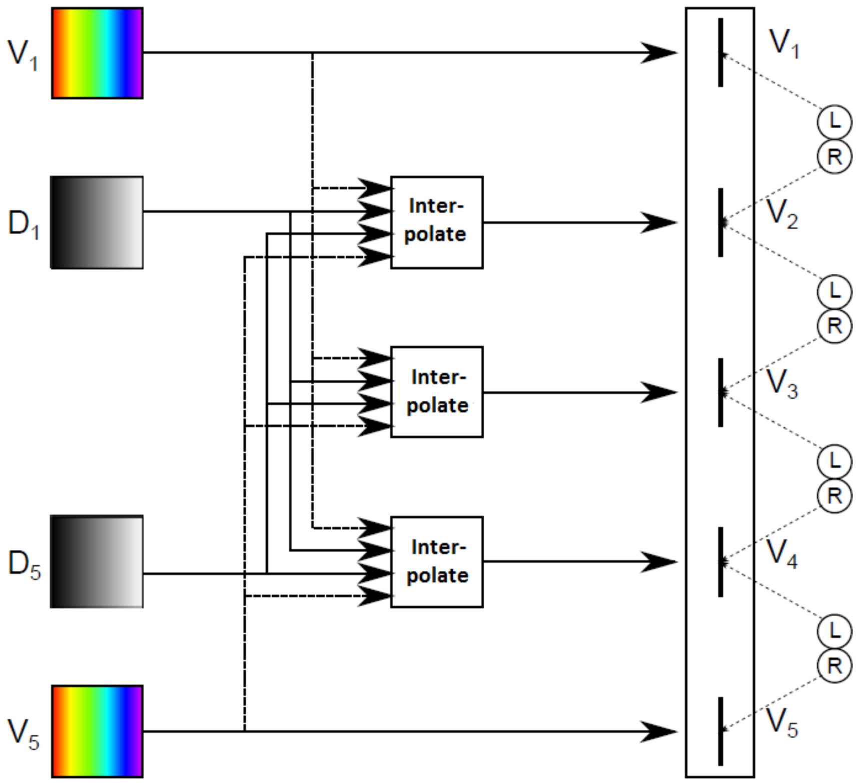


Figure 1.10 – *Intermediate synthesized views ($V2$, $V3$, $V4$) interpolated from main views ($V1$, $V5$).*

pendent views, same concepts and coding tools are used as for the base view. However, 3D specific tools are developed for coding more efficiently the dependent view by exploiting additionally the inter-view correlation. Major 3D tools will be described in the next sections.

1.3.2 Disparity-Compensated Prediction (DCP)

DCP (Muller et al. 2013) is an alternative concept to Inter prediction in 2D video coding which is also known as Motion-Compensated Prediction (MCP). In MCP, temporal redundancy between different pictures of a 2D video sequence is exploited. For 3D data, by extending MCP into the view direction, inter-view correlations can be exploited. DCP basically consists in adding an already decoded picture at the same time instant but in a different view in the reference picture list of the current frame. The MV found in this case does not represent motion as in 2D, but corresponds instead to a certain disparity between views and is thus called disparity vector (DV). A block coded in Inter mode and which is associated with a DV is said to be coded using DCP.

Figure 1.11 shows a current block in view V_k at time instant t_i coded in Inter mode, either with MCP (MV pointing to a decoded picture at the same view V_k but at a different time instant t_{i-1}) or with DCP (DV pointing to a decoded picture at the same time instant t_i but in a different view V_j).

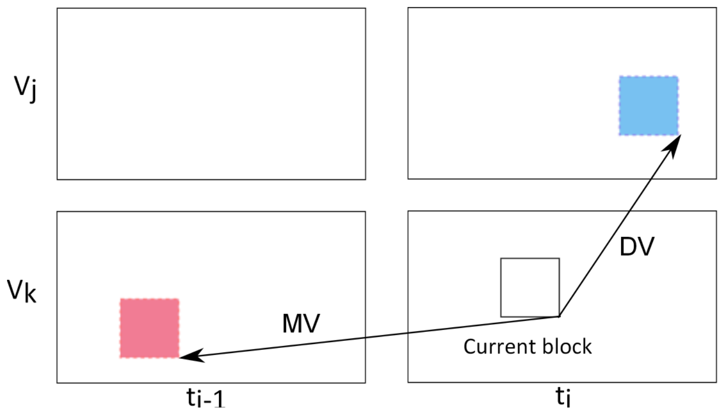


Figure 1.11 – *Motion-compensated (red) and Disparity-compensated predictions (blue).*

1.3.3 Inter-View Motion Prediction (IVMP)

IVMP (An et al. 2012, Zhang et al. 2014) is a method that provides improvement in coding efficiency of the dependent views. The basic concept of IVMP in 3D-HEVC is to derive the already coded motion parameter from the base view to be used in the dependent views. The reason behind this MV inheritance is that MVs in two different views are often correlated as shown in figure 1.12.

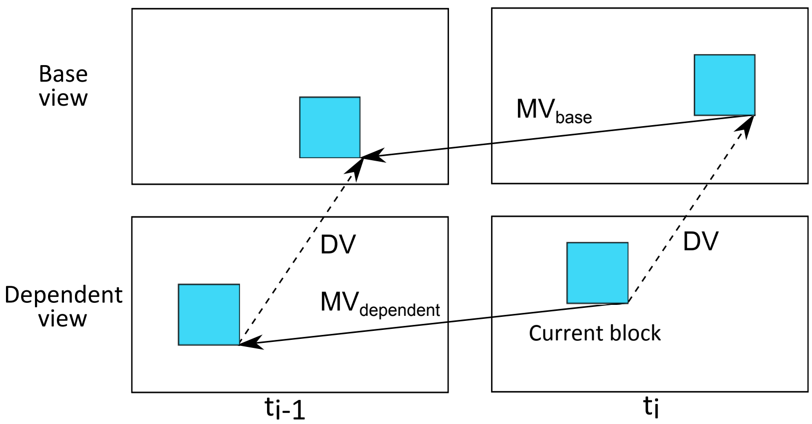


Figure 1.12 – *Correlation between motion vectors in base view and in dependent view.*

More precisely, when coding a current PU in a dependent view, a DV is first derived. By adding the DV to the middle position of the current PU, a reference sample location is obtained. The block in the already coded picture in base view that covers the sample location is used as the reference block. If this reference block is encoded using MCP, the associated MV can be inherited for the current PU in the current dependent view. Since that MV is not signaled in the bit stream, saving in signaling overhead is made.

1.3.4 Sub-PU Inter-View Motion Prediction (SP-IVMP)

SP-IVMP (An et al. 2013) is an improvement of IVMP and is thus a method to encode dependent views by inheriting motion parameters from the base view. In IVMP depicted in figure 1.12, the corresponding area in the base view pointed by the DV may have several different MVs (e.g. multiple objects inside the area); however, only the MV in the middle position of the area is inherited for the current PU in the dependent view. To improve this point, SP-IVMP proposes to split current PU in multiple smaller sub-PUs and to inherit motion parameters on sub-PU level.

More precisely, as illustrated in figure 1.13, the current PU to be encoded in a dependent view is divided into multiple sub-PUs of smaller size. For each sub-PU, a DV is added to the middle position pointing to a reference sample block in the base view. For each of those reference blocks, if it is coded using MCP, i.e. Inter mode, the associated MV is inherited for the corresponding sub-PU in the dependent view. Eventually, the motion parameter of the current PU is composed of multiple MVs corresponding to inner sub-PUs, all of them derived from the base view.

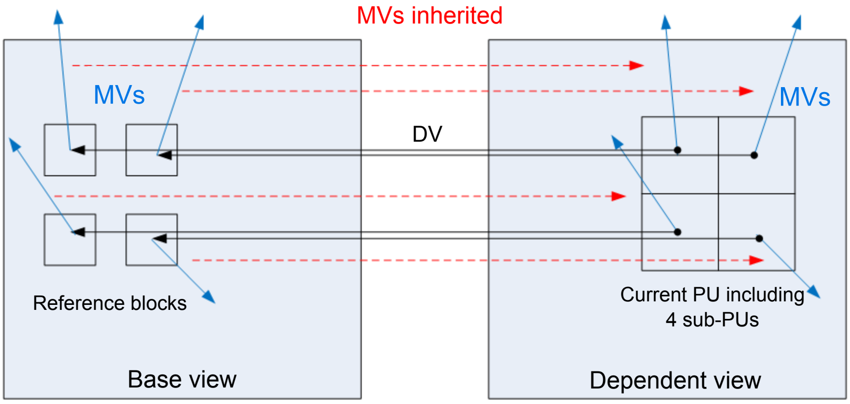


Figure 1.13 – *Inheritance of MVs on sub-PU level from base view to dependent view in SP-IVMP .*

By providing finer granularity for the MVs derivation process, SP-IVMP yields significant improvement in bitrate savings compared to IVMP method.

1.3.5 Advanced Residual Prediction (ARP)

ARP technique (Li et al. 2013) aims to further improve the coding efficiency of inter-view residual prediction, which predicts the residual of current block in the dependent views using the residual of its reference block pointed by the DV in the base view. The residual predictor is produced by aligning the motion information at the dependent view for motion compensation in the base view.

Figure 1.14 illustrates in more detail the ARP mechanism. Let us suppose that we are encoding the current block in a dependent view. After having its motion parameters (which point to pictures *Ref0* of the reference picture list*L0* and *Ref1* of *L1*), instead of computing directly its residual by applying motion compensation in the dependent view, a residual predictor is computed by referring to the base view: motion of the current block in the dependent view is applied to a reference block pointed by the DV in the base view (red arrows indicate the inheritance of motion parameters from the dependent view to the base view) in order to generate the residual (between the green block and the blue block) in the base view to be used as the residual predictor.

The reason behind this motion inheritance is to ensure a high correlation between residuals of the two views. Indeed, since the residual predictor in the base view and the residual to be predicted in the dependent view belong to the same object thanks to the motion inheritance, the prediction performance is improved.

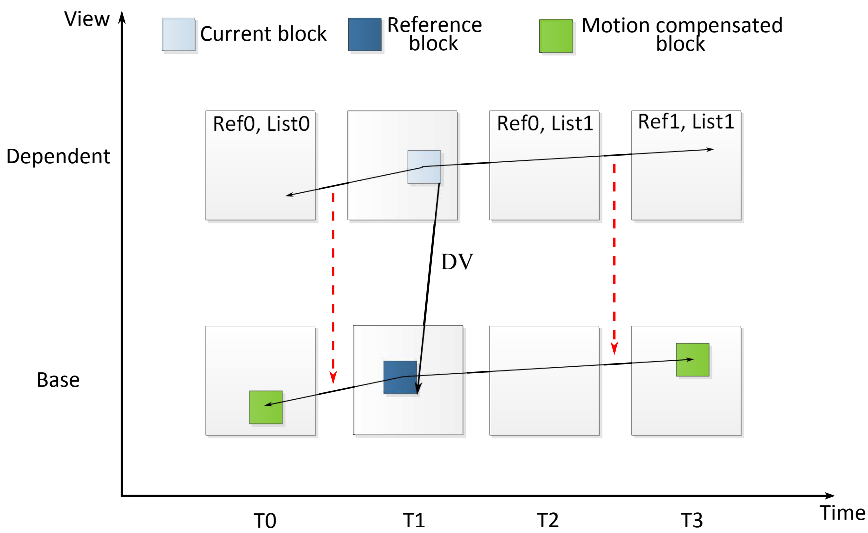


Figure 1.14 – *Inheritance of MVs (red arrows) to compute residual predictor in Advanced Residual Prediction.*

When ARP is enabled for one block, the difference between the current residual and the residual predictor is signaled. With the consideration of both temporal motion and disparity information, the residual predictor is derived more accurately. Moreover, adaptive weighting factors are applied on the residual signal to compensate the quality difference between views so that the prediction error is further reduced.

1.4 Video coding techniques exploiting a complex decoder

Up until now, there are interesting coding techniques which stand out from conventional approaches by proposing different frameworks that require significant modification at the decoder side. For better understanding, we will briefly explain the evolution of the decoder, from the most simple form that only reads data from the bit stream to complex structures able to simulate the encoder in order to make a decision.

1.4.1 Basic decoder to a more complex decoder with joint computation

In classic coding schemes, for each block, the encoder performs the R-D competition of all coding modes and transmits the best encoding option to the decoder. The latter then simply reads the encoding information in the bit stream and decodes a block accordingly. All the computation is concentrated at the encoder while the decoder is very light. The decoder can be considered in this case as *passive decoder*. A video codec scheme using this type of decoder is illustrated in figure 1.15, where all coding parameters, such as coding modes *param* and motion information $\overrightarrow{mv}$, must be transmitted in the bit stream to the decoder.

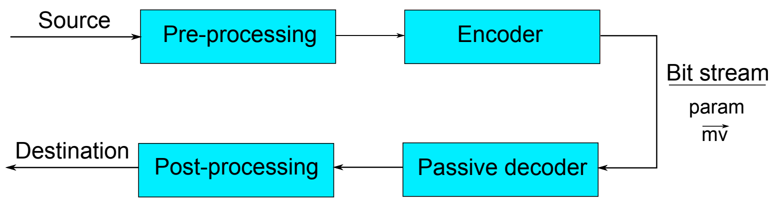


Figure 1.15 – *Video coding scheme with passive decoder.*

Various studies have emerged recently with the objective to further exploit the decoder by providing it with some processing ability. Joint computation in both encoder and decoder becomes possible, allowing some information to be retrieved in the decoder side without being signaled in the bit stream, reducing eventually the signaling overhead. Notable examples include approaches based on "Template Matching", a process that exploits the causal surrounding area of a block to derive motion parameters without signaling. This principle applied in Inter (Sugimoto et al. 2004, Kamp et al. 2008; 2009) proposes motion estimation that is jointly performed in both the encoder and the decoder. Resulting MVs are not transmitted, saving thus the signaling cost. Other variants can be applied in Intra (Tan et al. 2006) where the current block takes as predictor a block located in the reconstructed area of the current frame. In both cases, without receiving the MVs from the encoder, this type of decoder, called *active decoder*, can still retrieve those MVs by itself by performing motion estimation process similarly as in the encoder.

Figure 1.16 shows the modified codec scheme using an active decoder that incorporates some computation ability such as conducting the template matching process. Comparing with figure 1.15, we observe that there is no need to transmit the motion data $\overrightarrow{mv}$ in the bit stream to the decoder because the decoder can conduct the same template matching process as in the encoder side in order to compute $\overrightarrow{mv}$ by itself.

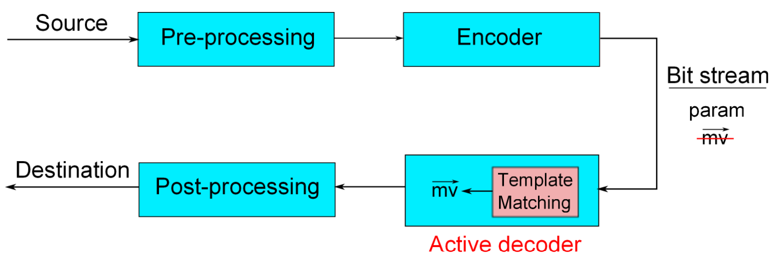


Figure 1.16 – *Video coding scheme with active decoder able to perform Template Matching process.*

1.4.2 Complex decoder with R-D competition

In (Thiesse 2012), a novel approach is proposed to further exploit the processing ability of the decoder. Not only simple computation processes such as template matching can be conducted as previously mentioned, the decoder is also given the ability to simulate the encoder by conducting the

R-D competition. Therefore, it is possible to perform the R-D competition of available coding modes on an already reconstructed block in an exactly identical way in both the encoder and the decoder. The coding mode and its associated parameter resulting from this competition do not need to be transmitted in the bit stream, as it can be recomputed by the decoder itself as illustrated in figure 1.17 depicting a coding scheme using the feature of a complex decoder. Consequently, the signaling overhead dedicated for coding modes and their parameters can be reduced.

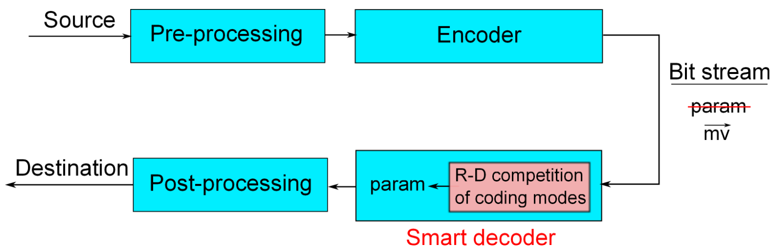


Figure 1.17 – *Video coding scheme with complex decoder able to perform R-D competition.*

This work is considered as a stepping stone for this thesis, where the complex decoder able to compute R-D competition becomes the main subject for further investigations in our research.

1.5 Improvement techniques related to our works

This section is dedicated to outline several techniques that aim to improve the video coding performance and are closely related to the research done during my thesis. First, Intra 1D is presented in the scope of 2D coding. Then Warped Optical Flow is described as an improvement for 3D coding. We provide finally an overview on machine learning based approaches that are recently proposed to be used in video coding.

1.5.1 Intra 1D

Intra 1D (Laroche et al. 2009) is a technique which aims to improve Intra coding mode. Conventional codecs use blocks square partitioning. As a result, there are more prediction errors for the bottom right part of the block than for the top left part due to greater spatial distance between pixels to be predicted and their reference pixels (located in the row above and the column on the left of the block). To solve this issue, Intra 1D proposes to use a 1-dimensional partitioning for the current block, where pixel to predict has closer neighboring reference pixels. By reducing the distance between pixels to be predicted and their predictors, the prediction accuracy is greatly improved.

Figure 1.18 illustrates different 1D partitionings for a block. They are completely defined by an initial 1D-partition p_0 transmitted to the decoder as an index for the Intra 1D mode. The other 1D-partitions are obtained by propagation of the initial pattern p_0 with mathematical morphology dilatation operation.

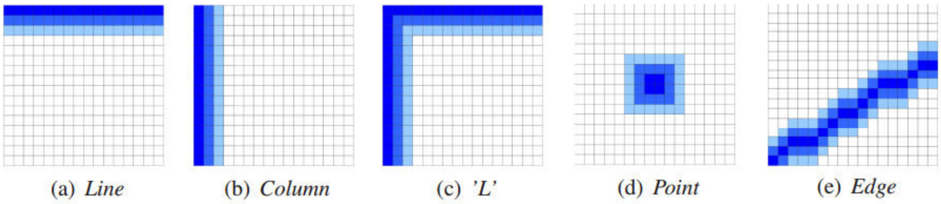


Figure 1.18 – *Different shapes of 1D partitions, obtained by propagating initial 1D-partition (in darkest color) (Source: (Thiesse et al. 2009)).*

Three scan orders are proposed to process 1D-partitions in a block as shown in figure 1.18: raster, bi-directional and hierarchical scans.

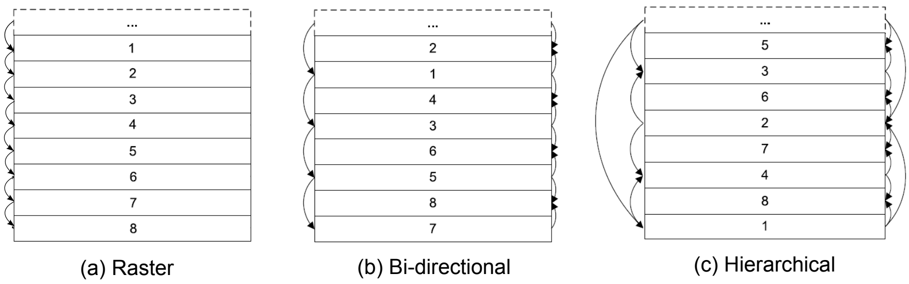


Figure 1.19 – *Different scan orders for 1D partitions in block (Source: (Laroche et al. 2009)).*

The reason behind the use of the bi-directional scan is that it appears sometimes more efficient in terms of R-D criterion to encode a partition with a farther reference partition and then encode the remaining partition with more adjacent reference partitions, generating thus higher spatial correlations. For the hierarchical scan, it is inspired from the temporal pyramidal order of processing P- and B-frames when encoding a sequence.

Different predictors are defined specifically for each of three scan orders, for example using the previous line, the left pixel of current 1D-partition, etc. Thanks to the multitude of defined predictors, each 1D-partition can be predicted efficiently. 1D DCT transform, instead of standard square DCT transform, is used to encode the residual of a 1D-partition.

1.5.2 Warped Optical Flow (WOF)

Basically, optical flow (OF) is a method for motion estimation. Unlike block matching algorithm that produces only one MV for the whole block, OF derives a *Dense Motion Vector Field* (DMVF) where each pixel is associated to a MV. Many OF algorithms have been developed, prominently the Horn and Schunck (Horn and Schunck 1981) and Lucas and Kanade (Lucas and Kanade 1981) methods. By using a finer granularity for deriving MVs, a better motion prediction can be obtained with OF. However, most approaches involve a direct use of the DMVF to encode the motion parameters (Dufaux and Moscheni 1995, Krishnamurthy et al. 1995, Shi et al. 1997) and suffer from several problems, especially the fact that there is a MV for every pixel, which increases the amount of motion parameters for transmission as well as being completely incompatible with any established standard.

Unlike other OF based approaches, WOF (Mora 2014) is a technique that proposes a new way to integrate OF into video coding. As depicted in figure 1.20, it extends the IVMP approach such that a DMVF in the base view is computed using OF technique based on already reconstructed frames and will be used later for inter-view prediction purpose to improve dependent views. The basic idea is to additionally perform a computation of OF at decoder side, while DMVFs are strategically computed between already reconstructed frames in the base view and are inherited for dependent views similar as IVMP/SP-IVMP technique. DMVFs can be therefore recomputed at the decoder exactly the same way as in the encoder, removing the need to transmit all the MVs for every pixel.

When encoding a PU in a dependent view, each pixel inherits a MV from the DMVF of corresponding PU in base view pointed by the DV, hence increasing the prediction accuracy and reducing the residual energy. Indeed, it is proved in (Mora 2014) that the DMVF computed using OF based on two already reconstructed frames, although suffering from the quantization noise, still gives more accurate frame predictions than the coarse block-based MV.

Similar to IVMP/SP-IVMP, the inherited DMVF is introduced as a new candidate in the Merge candidates list and is signaled by an index for each PU. Aside, there is no additional cost for signaling the DMVF since it can be recomputed at the decoder side based on already decoded frames in the base view. The decoder in this case is required to perform additionally the computation of OF, which in turn significantly increases the runtime.

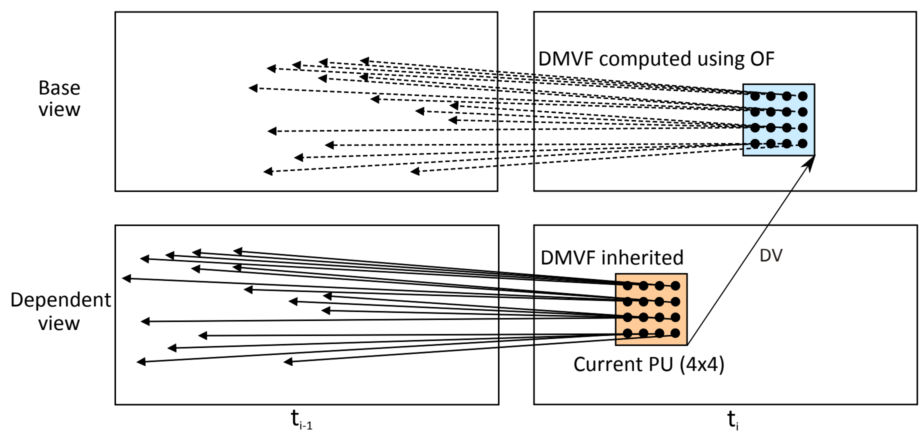


Figure 1.20 – *DVMF computed using OF based on reconstructed frames in base view at instant t_i and t_{i-1} is inherited to encode a PU in dependent view at instant t_i in Warped Optical Flow technique.*

1.5.3 Machine learning based approaches

Machine learning is a scientific discipline that constructs algorithms which can learn from past data and then make predictions rather than only following explicitly programmed instructions. For better understanding of our work presented later in this manuscript, this section first introduces an overview about machine learning, then presents a state of the art on existing approaches that propose to exploit machine learning in video coding.

1.5.3.1 Overview on machine learning

In general, machine learning aims to discover and learn properties from a known data set (i.e. *training set*) by building a model based on already known information, then tries predicting an unknown data set (i.e. *test set*).

We can separate learning problems into two main categories:

- *Supervised learning*, where the objective is to infer a function from labeled training data. Each data sample is a pair consisting of its attributes (i.e. *features*) and its label (i.e. *supervisory signal*). A supervised learning algorithm analyzes training data samples and produces an inferred function (i.e. *classifier*), which can be used for mapping new samples whose label is unknown. Several algorithms used to construct the classifier exist, such as decision trees, K-nearest neighbors or Support Vector Machine (SVM) (Cortes and Vapnik 1995). Figure 1.21 shows an example of binary classification using SVM, where an optimal hyperplane separating two classes is computed by solving the margin maximization equation.
- *Unsupervised learning*, in which the training data consists of a set of samples without any corresponding label. Unsupervised learning aims to discover groups having similar properties within the data (i.e. clustering), or to determine the distribution of data within the input space (i.e. density estimation).

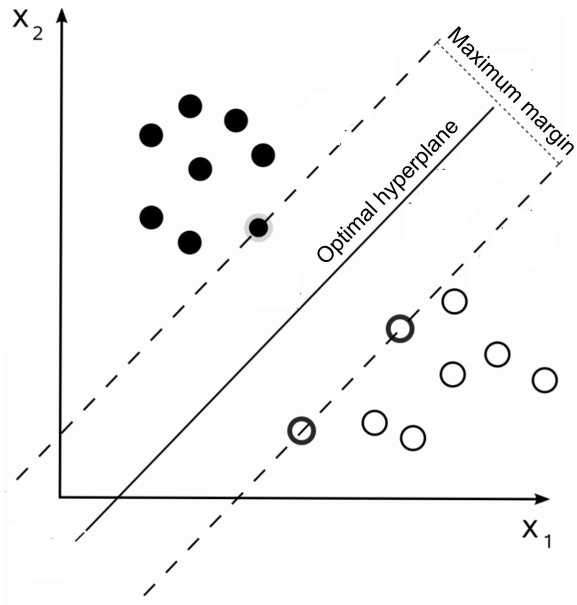
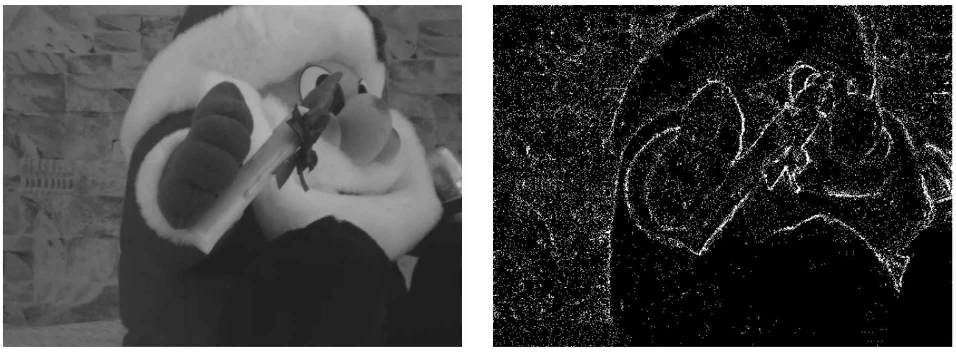


Figure 1.21 – *Maximum-margin hyperplane and margins for an SVM trained with samples from two classes.*

1.5.3.2 Video coding techniques based on machine learning

Machine learning has been widely used in image and video processing for applications such as content based image and video retrieval, content understanding, and more recently video mining. However, only a few concern video coding. The idea behind using machine learning in video coding is to exploit structural similarities in video in order to make an accurate prediction of the optimal coding mode of a block.

In (Cheng and Vishwanathan 2007, Bai et al. 2009), authors present a novel learning-based video compression framework. Unlike conventional block based coding schemes, proposed approach compresses the texture image on the frame level, by choosing a small number of representative pixels (RP) at the encoder to be signaled in the bit stream instead of performing conventional frequency transformation. A semi-supervised learning process is then performed at the decoder to reconstruct the entire image by predicting the rest of pixels based on signaled RPs as illustrated in 1.22. However, initial results show that the scheme has a poor performance compared with existing codecs.



(a) Decoded Y frame (b) Representative pixels

Figure 1.22 – *Compressing entire luminance frame (a) using learning method based on representative pixels (b) (Source: (Bai et al. 2009)).*

Other approaches use machine learning techniques mainly to reduce encoding runtime, by creating learning-based fast mode decisions. Indeed, block mode decision is computationally the most expensive process due to the exhaustive R-D competition of all available options, such as block coding mode, block partition, Intra directions or Inter motion estimation. Therefore, machine learning is exploited to give relatively accurate decisions on some block parameters, allowing to skip the time consuming R-D competition. Different classification algorithms are proposed as follows:

- In (Han et al. 2010, Kalva and Christodoulou 2007, Jillani and Kalva 2009, Carrillo et al. 2010, Ma et al. 2009), decision trees are used to find decision rules on the optimal partition size or the optimal Intra direction (fig. 1.23). Predetermined features, such as the mean and variance values of the current block, are used for the classification process.
- In (Lampert 2006, Tohidypour et al. 2014, Zhou et al. 2009), the authors propose fast mode decision methods based on Bayesian classifier to predict block parameters using the information of already encoded neighboring blocks. In (Di and Yuan 2010), K-means clus-

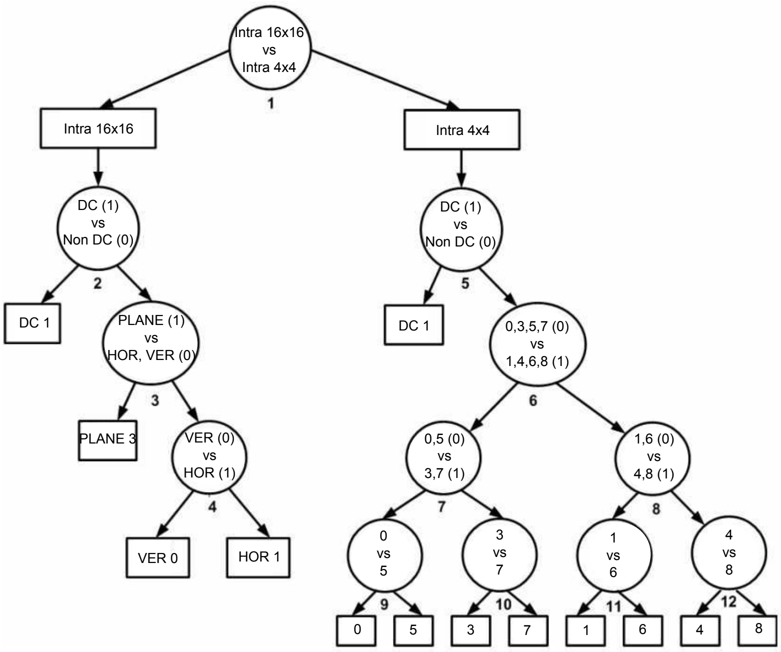


Figure 1.23 – *Decision tree for Intra block partitions and directions.*

tering is used as the prediction tool for the proposed early termination strategy. In (Chiang et al. 2011), the authors exploit SVM to determine the best decision from the extracted features in order to reduce the computational cost involved in three main components of the encoder: prediction of partition size in Inter mode, motion estimation and Intra mode prediction.

– In (Xiong and Li 2012), it is proposed to use non-normalized histogram of oriented gradient (n-HOG) as a criterion for the fast mode decision. The authors prove that n-HOG can be used to decide whether or not the block should be further split. More precisely, for each block size, n-HOGs of training sequences are clustered to construct a codebook offline. The decision to split the current block is determined by comparing the n-HOG of the block with the codebook, rather than by conducting a full R-D check, reducing consequently the runtime.

All previously mentioned machine learning based fast mode decisions are able to efficiently reduce the encoding time while limiting the degradation in compressed video quality to an acceptable level.

According to our understanding, there is still no efficient approach that successfully exploits machine learning for the objective of increasing compression ratio, i.e. reducing coded bits transmitted in the bit stream.

Conclusion

In this chapter, an overview of the state-of-the-art HEVC and 3D-HEVC standards is given. Major tools related to our works are further detailed. Compared to previous video coding standards, HEVC features several incremental improvements which provide its compression performance, in particular an increased number of competing coding modes and parameters. This tendency could however have some limits, such as constraints generated from signaling overhead, that we will refer in more detail in the next chapter. 3D-HEVC, aside from including all HEVC features, furthermore provides 3D specific coding tools that exploits inter-view correlation to encode dependent views. The state of the art on several proposed improvements for both 2D and 3D is also presented, such as Intra 1D mode, WOF, techniques exploiting complex decoder, and maching learning based coding techniques. All of these improvements provide the background for our research.

The knowledge about HEVC and 3D-HEVC is required for further readings, as next chapters will present approaches that concentrate on different specific part of the encoding process.

Part 1

SDec scheme and its practical applications

2 Video coding scheme based on Smart Decoder (SDec)

As more coding modes are introduced to further exploit different video signal correlations, the signaling cost resulting from the competition of those coding modes is greatly increased and generates significant overhead. In this chapter, to anticipate the limits of conventional video coding schemes and further improve the coding efficiency, we propose a solution based on the competition of multiple coding choices. It is based on the use of the "Smart Decoder" (SDec), referring to a complex decoder able to reproduce the competition of coding modes and parameters so that their signaling cost can be reduced, improving thus the coding efficiency.

Our motivation is first explained. The general outline of the proposed encoding and decoding SDec scheme is then presented. Next, advantages and drawbacks of SDec scheme are described. Finally, further analysis on some particular characteristics of the SDec scheme is provided.

2.1 Motivation

The compression ability of HEVC comes from the use of Inter and Intra predictions that exploit redundancies in time and space, respectively. As already mentioned in the chapter 1, compared with the previous standards, HEVC features an increased number of competing coding modes and their associated parameters (ITU-T 2013), allowing an improvement in prediction accuracy for both motion and texture data.

In (Laroche 2009), the proportion of different syntax elements in the total bit stream when coding a P-frame in H. 264/AVC is analyzed, as depicted in figure 2.1. We observe that apart from the texture residual, signaling the motion parameters for a P-frame costs a significant portion of the total bit stream. Therefore, several works have been done after the standardization of H.264/AVC in order to improve the coding efficiency of those syntax elements. Most of them, including the MV-Comp technique (Jung and Laroche 2006) which is considered as the predecessor of AMVP, can be summarized in (Thiesse 2012).

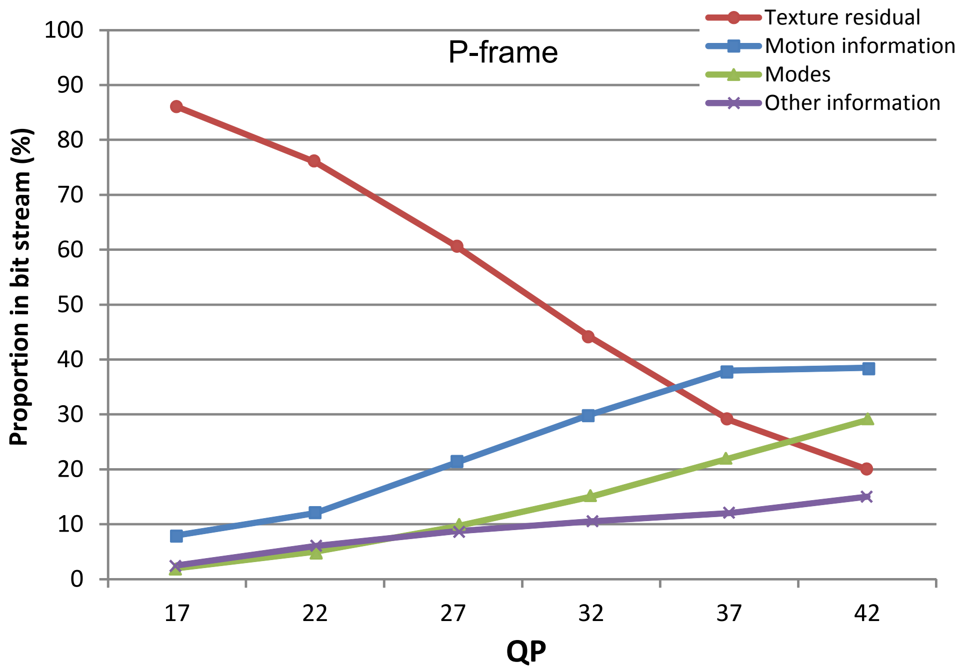


Figure 2.1 – *Proportion in bit stream for different syntax elements in H.264/AVC.*

After conducting similar analysis, by observing the proportion of different syntax elements in the total bit stream in HEVC as depicted in figure 2.2, we can conclude that thanks to the effort made which aims to improve the efficiency of motion prediction, the signaling cost of motion information (blue line) is indeed reduced, down to less than 20% of the total bit stream for a normal QP range 22-37.

By comparing the distribution of syntax elements for P-frame in both figures above, we can notice that the signaling cost for coding modes (green line) is increased significantly from H.264/AVC to HEVC, in contrast to other components, due to the introduction of new coding modes

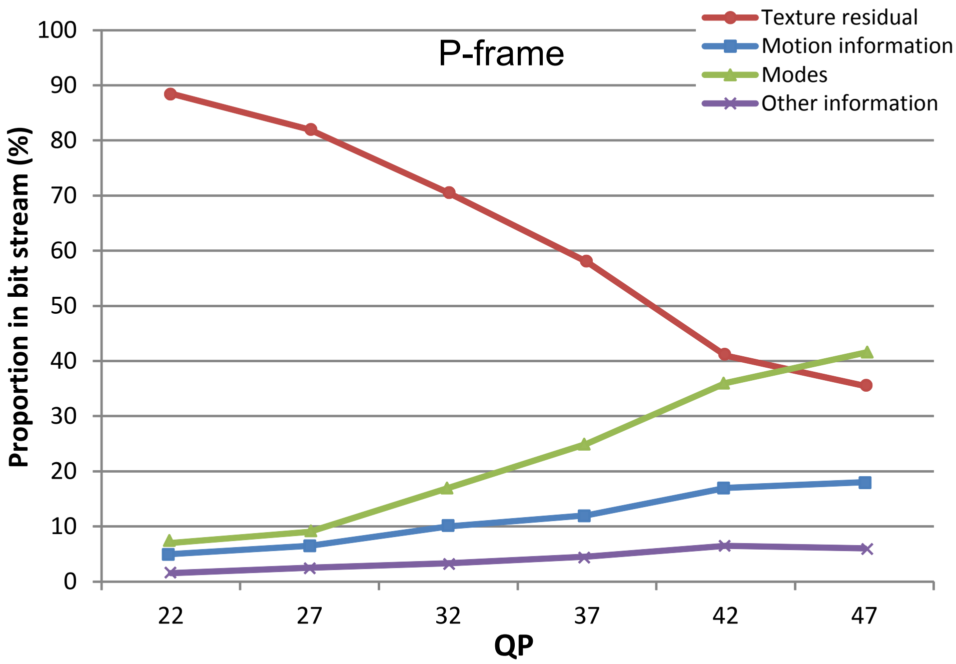


Figure 2.2 – *Proportion in bit stream for different syntax elements in HEVC.*

and parameters which allow however a more accurate prediction.

In the future, more coding modes could be added in order to better exploit signal correlation and consequently generate an unavoidable overhead that could limit the compression performance. Therefore, it is important to find a way to reduce the signaling of coding modes so that multiple coding modes can be integrated without significant impact on the transmitted bit stream.

2.2 General description of SDec based video coding scheme

Given that reducing the signaling overhead is crucial in modern video coding, we propose a new coding scheme which aims to reduce the signaling of coding modes and their associated parameters, since their numbers are expected to be further increased in the future post-HEVC codec generations.

The idea originates from (Thiesse 2012), which consists in performing the computation for the optimal coding mode on a causal reference and then using that mode to encode the current block, as depicted in figure 2.3. Similar to the encoder, the decoder is provided with the ability to perform similar process on the same causal reference. Thus the decoder can retrieve the exact optimal coding modes and parameters selected at the encoder side and use them for decoding the current block. Those coding modes do not need to be signaled in the bit stream, reducing in consequence the signaling overhead. This new approach can be classified in the same category as approaches that exploit a complex decoder to improve the coding performance already mentioned in the state of the art. The difference is that the decoder in our approach has a higher level of complexity because it can simulate the encoder by conducting for example identical rate-distortion (R-D) competition. Such a decoder is therefore called "smart decoder".

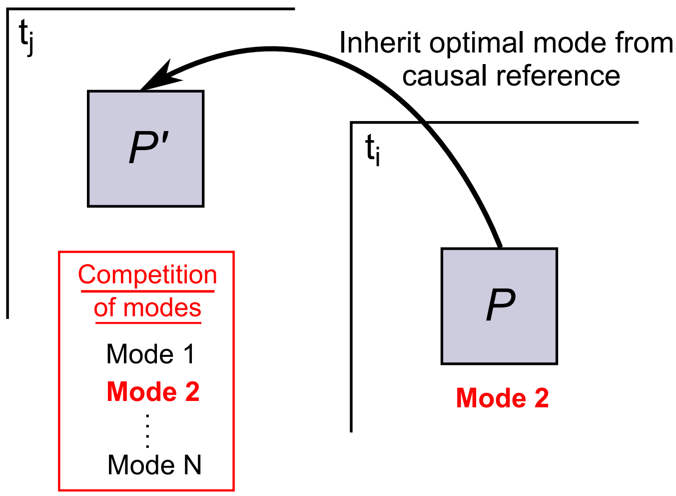


Figure 2.3 – *Basic idea of the SDec scheme: use the optimal coding mode computed on a causal reference P' located in an already decoded frame at instant t_j to encode the current block P located in the frame at instant t_i.*

We will use the following notations in the encoding and decoding processes of the general SDec scheme, which are presented separately in the next sections:

- P the current PU to encode,
- P' the SDec PU reference selected among n candidates $(P'_1, ..., P'_n)$,
- $(M_1, ..., M_m)$ m coding modes, each having associated parameters $(S_1^{M_i}, ..., S_q^{M_i})$. For example, M_1 could be HEVC Intra mode, with $(S_1^{M_1}, ..., S_q^{M_1})$ corresponding to different Intra directions: horizontal, vertical, DC, etc.
- M_P^* and S_P^* the optimal coding mode and parameter selected by SDec for encoding P,
- $Pred_{M_P^*, S_P^*}(P)$ the prediction of P using M_P^* and S_P^*,
- $\varepsilon_T(P)$ the texture residual after encoding P.

Moreover, two syntax elements are introduced, which are described as follows:

- $sdec\_ref$ the syntax element signaling P',
- $sdec\_flag$ the syntax element indicating whether or not the SDec mode is used to encode P.

2.2.1 Encoding scheme

In conventional encoding scheme, all available coding modes with their associated parameters are set to compete on the current PU. The optimal mode which minimizes the R-D cost is then selected to encode the current PU and is signaled in the bit stream. In the proposed encoding scheme, the selected coding mode is not calculated directly on the current PU to encode, but instead on a causal PU called the SDec reference PU.

Figure 2.4 summarizes different steps of SDec encoding scheme, which encodes the current block using the new SDec mode being introduced to compete with other existing coding modes. The whole process can be described in four steps:

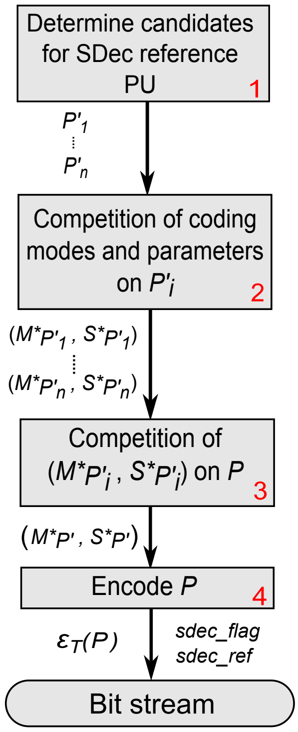


Figure 2.4 – *General outline of the SDec encoding scheme.*

Step 1: Determine the candidates for the SDec reference PU P'
In the general SDec scheme, three approaches are possible for selecting P':
- P' is chosen from a list of pre-identified PU candidates. $sdec\_ref$ is then the index of the selected candidate and is transmitted.
- P' is dynamically calculated using a motion estimation. $sdec\_ref$ is then the motion vector pointing to P' and is transmitted.
- P' is a pre-identified PU (such as the colocated block of P). There is no need to transmit $sdec\_ref$.

In a specific implementation of the SDec scheme, one of these approaches, known by the decoder, is used for all PUs.

Step 2: Competition of coding modes and parameters on P'_i
This step computes the optimal coding mode and parameter for each SDec reference PU P'_i. All coding modes and parameters are competing to encode P'_i. More specifically, each pair $(M^j, S_q^{M_j})_{j\in[1..m],l\in[1..q_j]}$ of coding mode and parameter is used to encode P'_i in order to evaluate the optimal pair in terms of R-D cost:

$$J = D + \lambda R$$

where J is the R-D cost, D is the distortion between the reconstructed and the original PU (which correspond respectively to the PU obtained from the encoding of P'_i, and P'_i), R is the estimated encoding rate, and λ is the Lagrange multiplier that depends on the quantization parameter. Since the coding modes and their associated parameters are not transmitted in the SDec scheme, R does not include their signaling cost. The R-D computation requires therefore only the distortion D which is computed from the texture residual. This residual results from each encoding of P'_i using $(M^j, S_l^{M_j})$ and is calculated as follows:

$$\varepsilon_T(P'_i) = P'_i - Pred_{M_j, S_l^{M_j}}(P'_i)$$

At the end of this step, each of all n candidates P'_i has a pair of optimal coding mode and parameter, noted $(M^*_{P'_i}, S^*_{P'_i})$, which minimizes J.

Step 3: Competition of $(M^*_{P'_i}, S^*_{P'_i})$ on P
This step determines, among n optimal coding modes corresponding to n candidates P'_i, the optimal coding mode for the current PU P using R-D based competition. Each optimal pair of coding mode and associated parameter $(M^*_{P'_i}, S^*_{P'_i})_{i\in[1..n]}$ is used to encode P and the optimal pair $(M^*_{P'}, S^*_{P'})$ is selected, with P' the associated SDec reference PU.

Note that this step is conducted separately from the step 2 to guarantee the decodability of the encoded sequence. Indeed, unlike the second step, this third step is not causal, i.e. not reproducible at the decoder side, because the competition is conducted on P. This step is skipped if there is only one candidate for P'.

Step 4: Encode P
This step encodes P using the optimal coding mode and parameter $(M^*_{P'}, S^*_{P'})$. A texture residual resulting from the encoding of P is finally calculated to indicate the prediction error compared to the original PU:

$$\varepsilon_T(P) = P - Pred_{M^*_{P'}, S^*_{P'}}(P)$$

Regarding the information signaled in the bit stream, the syntax element $sdec\_flag$ signaling the use of SDec mode among available coding modes (Skip, Inter, Merge, Intra...) is transmitted, along with the syntax element $sdec\_ref$ indicating which P' is selected. The coding mode and its associated parameter used to encode the current PU are indeed not transmitted. This constitutes an important characteristic of the proposed scheme.

2.2.2 Decoding scheme

At the decoder side, the coding mode of the current block is parsed from the bit stream, then the reconstructing process corresponding to this coding mode is called to rebuild the block. In case of the SDec mode, i.e. $sdec\_flag = 1$, we invoke the SDec decoding scheme which consists of three different steps as shown in figure 2.5.

Step 1: Parsing for the decoding information
The bit stream is parsed to retrieve the data necessary for the SDec decoding process, for example the $sdec\_ref$ syntax element needed to determine P'.

Step 2: Competition of coding modes and parameters on P'
The competition of all coding modes and their associated parameters is performed on P'. Simulating the encoder, the decoder encodes P' to find the pair $(M^*_{P'}, S^*_{P'})$ that minimizes the R-D criterion. Remark that this step is exactly the same as the second step previously described in the encoding scheme, so that the resulted pair of optimal mode and parameter

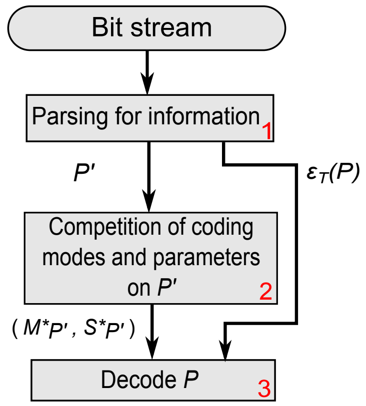


Figure 2.5 – *General outline of the SDec decoding scheme.*

$(M^*_{P'}, S^*_{P'})$ can be computed identically in both the encoder and the decoder sides.

<u>**Step 3**</u>: Decode P
The current PU is decoded using the optimal coding mode and parameter $(M^*_{P'}, S^*_{P'})$. The texture residual $\varepsilon_t(P)$ extracted from the bit stream is finally added to reconstruct P.

2.3 Advantages and drawbacks of the SDec design

The ability to have an unlimited number of coding modes and parameters in competition without suffering from the excessive signaling cost is a major advantage of the SDec scheme. Very different coding modes with their associated parameters can be set in competition, and consequently the adaptation to specific content or characteristic of the scene can be automatically handled. The SDec scheme consequently allows to integrate coding techniques that usually fail due to excessive burden resulting from the large number of their intrinsic parameters in competition, such as "compressed sensing" (Do et al. 2010) which requires to transmit transforms and sampling factors. A theoretical study and preliminary experiments are conducted in (Thiesse 2012), showing promising results (0.5% of gain in average) despite the significant complexity introduced. "1D short distance Intra prediction" (Intra 1D) (Cao et al. 2013, Thiesse et al. 2009) is another coding technique that could benefit from the use of the SDec scheme since its powerful capacity of prediction requires a lot of parameters to be signaled, which in turn limits its performance. The integration of Intra 1D in the SDec scheme will be presented later in the chapter 4 of this manuscript. Techniques based on "geometry partitioning" (Escoda et al. 2007) are also very good examples of possible applications for the SDec scheme. They aim to improve the prediction effiency by using a partitioning that is adaptively computed depending on the texture content of the block, thus requiring the transmission of several parameters to describe partitions boundaries. Using SDec scheme will help to reduce this signaling overhead by inheriting the optimal partitioning which is computed on the SDec reference and applying it on the current block to be encoded.

Furthermore, complex processes such as machine learning become applicable, thanks to the new ability to simulate the encoder at decoder side offered by the SDec scheme. Classification techniques, for example, can be used to establish a decision rule, based on causal blocks, which will be applied to encode the current block without being signaled in the bit stream. The decoder conducts the same classification processes to retrieve the choice made by the encoder. With the SDec decoder, this learning based decision can be performed on the fly during the decoding process. The exploitation of machine learning techniques in video coding will be presented in the chapter 8 of this manuscript.

The major drawback of the proposed scheme is the additional complexity at both the encoder and the decoder given that all coding modes and parameters must be tested on the SDec reference to find the choice to be inherited on the current block. Indeed, while conventional decoder only parses the bit stream for optimal coding mode and parameters then reconstructs the block accordingly, the SDec decoder must conduct the whole R-D competition exactly the same way as the encoder to retrieve the necessary information. However, this complexity is fully scalable, in the sense that processing power required to perform a task is allocated according to its need by the codec device. It can be negotiated and adapted on the fly in case of interactive applications or managed by profiles definition.

Another drawback is the sub-optimality of the coding mode that is inherited from the SDec reference to encode the current block in case both the SDec reference and the current block do not yield the same mode. Indeed, since there is no real competition on the current block, the inherited coding mode cannot be considered as truly optimal, as opposed to the mode computed directly on the current block. This drawback can be minimized if there is a significant similitude or correlation between the SDec reference and the current block, leading to the problematic of selecting blocks for the SDec reference which will be further discussed in the next section.

Moreover, the design of the SDec scheme requires the SDec module to be implemented, if at the decoder side, in the parsing phase where block information is extracted from the bit stream. As such, data provided by the SDec process can be exploited to parse necessary information for block decoding. This prevent the independent parsing of the bit stream and makes the parallel computing more difficult.

2.4 Selection of the SDec reference

In the SDec scheme, the SDec reference plays a very important role because it determines the coding modes and parameters that are used to encode the current block. Therefore, the similarity between both the SDec reference and the current block guarantees that the optimal coding mode computed on the SDec reference is also optimal for the current block, improving obviously the efficiency of the SDec scheme.

In the illustrating example shown in figure 2.6, we suppose that available coding modes consist of only 35 Intra directions. Being very similar to the current block P, P_1' is a better choice for the SDec reference compared to P_2'. Indeed, the optimal Intra vertical direction computed on P_1' is also the optimal Intra direction on P. Inheriting that Intra vertical direction mode using the SDec scheme to encode P is therefore relevant.

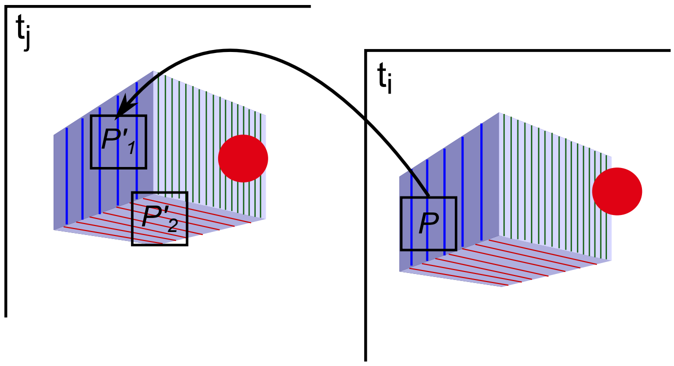


Figure 2.6 – *Example of efficiently selecting blocks to be the SDec reference: being more similar to P than P'_2, P'_1 is a better choice.*

However, it is not required that the SDec reference is identical to the current block. Indeed, since the basis of the SDec scheme is to encode the current block with a coding mode inherited from the SDec reference, the reference block should only be such that its optimal mode is also optimal on the current block. In other terms, to be a good SDec reference, it is enough for a block to have some similar characteristics as the current block regarding the optimal coding mode.

Let us consider again the previous example, with P'_1 and P'_2 as depicted in figure 2.7. Since we consider only the Intra mode with its 35 Intra directions during the competition for the optimal SDec coding mode, a good reference SDec P' does not have to be exactly identical to the current block P, but should only have the texture with the same direction as the texture of P. Therefore, although P'_2 is less similar to P when compared with P'_1, P'_2 is as efficient as P'_1 when being considered as the SDec reference because both of them yield the same optimal Intra vertical direction.

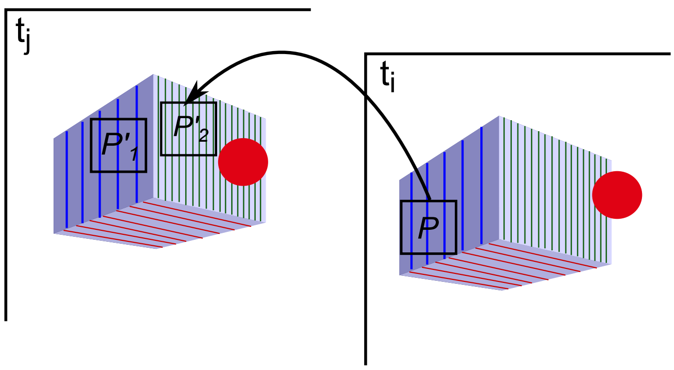


Figure 2.7 – *Example of efficiently selecting blocks to be the SDec reference: despite being less similar to P than P'_1, P'_2 is as efficient as P'_1 since they both yield the same optimal Intra vertical direction.*

In another example illustrated in figure 2.8, let us suppose that the available coding modes are limited to only DC and Planar Intra mode. Since DC is good to predict uniform block with smooth texture while Planar better predicts block with complex texture using linear interpolation, it could be more interesting to consider the transform domain rather than the pixel domain to evaluate the correlation between the SDec reference and the current block. Indeed, while both P_1' and P_2' are not similar to P in the pixel domain (P_1' has a more significant gray level than P and P_2' has texture different than P), P_1' is a good SDec reference because in transform domain, both P_1' and P have a similar spectrum which contains only a DC coefficient due to their uniform texture. In the end, both the SDec reference and the current block would have the DC as the optimal Intra coding mode, resulting in a good performance of the SDec scheme.

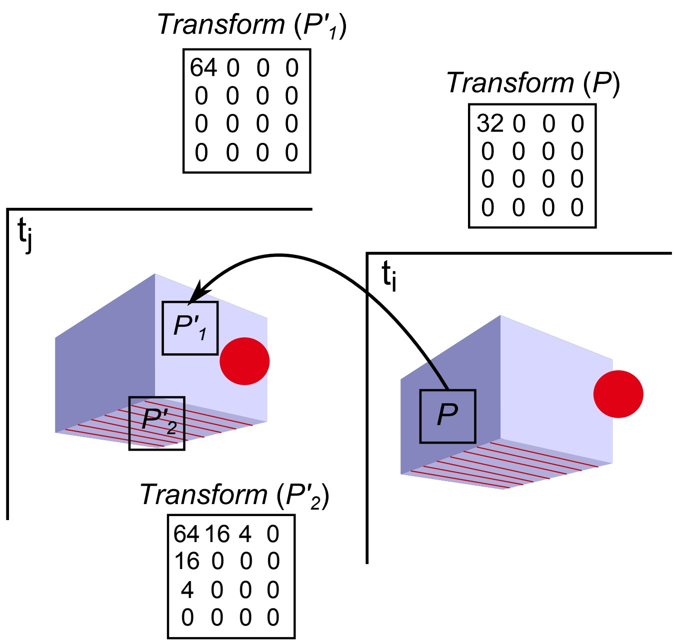


Figure 2.8 – *Example of efficiently selecting blocks to be the SDec reference: being more similar to P than P_2' in transform domain, P_1' is the better choice.*

Additionally, we can cite for example the technique using depth modeling modes (DMM), which is a depth coding method for multi-view plus depth sequences. Depicted in figure 2.9, this approach uses wedgelet- and contour-based depth modeling modes which are specifically adapted to the characteristics of depth maps, allowing thus a close approximation of a depth block by non-rectangular partitions.

This technique can be applied in the SDec scheme where the block partitioning (wedgelet or contour) is calculated on the SDec reference and is inherited to encode the current block, allowing thus to remove its costly signaling data. This particular block partitioning, applied on the current block, can help improving the encoding performance compared to the conventional quad-tree partitioning by providing more efficient prediction without over-partitioning the block. Given the nature of this method, a block could only have a texture partitioning similar as the partitioning of the current block to be considered as an efficient SDec reference.

In conclusion, an efficient SDec reference does not have to be exactly identical to the current block, but should only yield the same optimal coding mode. The tie-in between the selection of the SDec reference and the nature of a coding mode is thus highlighted, introducing the possibility of

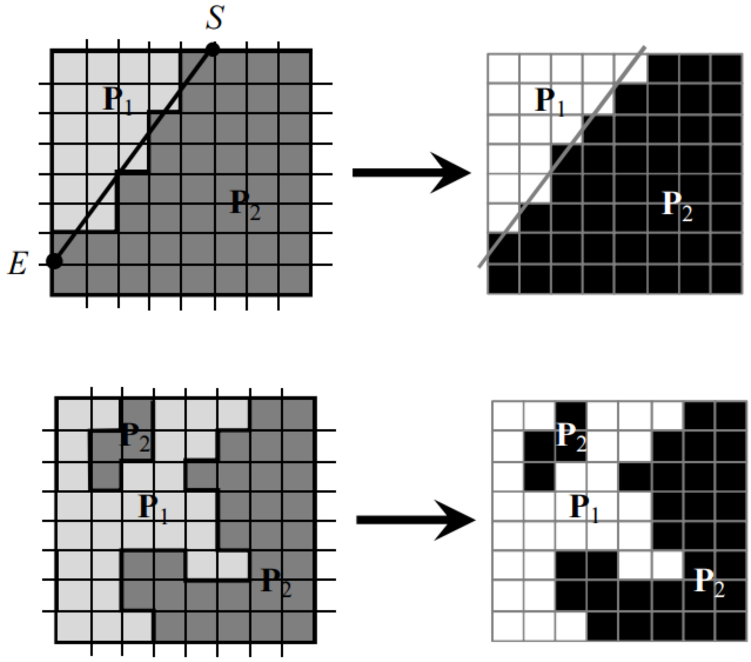


Figure 2.9 – *Wedgelet partition (top) and contour partition (bottom) of a depth block: original sample assignment to partitions P1 and P2 (left) and partition pattern (right) (Source: (Muller et al. 2012)).*

using different SDec references for different available coding modes. Indeed, different coding modes have different degrees of prediction for the current block. For example, the prediction accuracy of Intra with 1D partitioning is better than of HEVC Intra due to its reduced distance between pixels to be predicted and their pixel of reference. Therefore, for coding modes with high prediction accuracy, such as Intra 1D, to be used among the SDec coding modes, the SDec reference is better to be as similar to P as possible. On the contrary, for coding modes with lesser degree of prediction precision such as HEVC Intra mode, the constraint of similarity on the SDec reference should be less significant, allowing therefore to have an efficient SDec reference from blocks that do not ressemble much P. The flexibility to select the SDec reference depending on the coding modes being considered is thus an interesting point to note for the SDec scheme, giving a perspective for the selection of the SDec reference where the choice is made according to the degree of similarity between a block candidate and the current block. As each coding mode requires a different criterion when evaluating blocks to be the SDec reference, an adaptive process that selects block depending on the coding mode being considered can improve the coding performance. For example, for coding modes such as Intra 1D where the SDec reference block should be very similar to the current block, template matching technique can be used to identify the block instead of simply using a pre-identified block.

2.5 Partial standardization of the encoder using the SDec scheme

In standardization, only the bit stream syntax and the decoder are specified. Video coding standards are thus flexible enough to allow the competition between different manufacturers based on technical merit to optimize and to reduce the complexity for implementability, while guaranteeing the interoperability between devices. Figure 2.10 depicts the scope of video coding standardization.

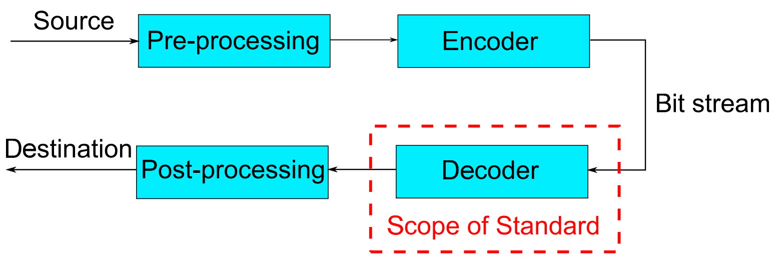


Figure 2.10 – *Scope of video coding standardization*

The SDec scheme requires the decoder to simulate the competition of coding modes at the encoder, so that the choice can be deduced at the decoder without being signaled in the bit stream. The whole SDec competition must be reproduced identically in both encoder and decoder sides to ensure the decodability. As a result, aside from the decoder which is already in the scope of the standardization, the encoder must also be partially standardized in order to be compatible with the SDec scheme, as depicted in figure 2.11.

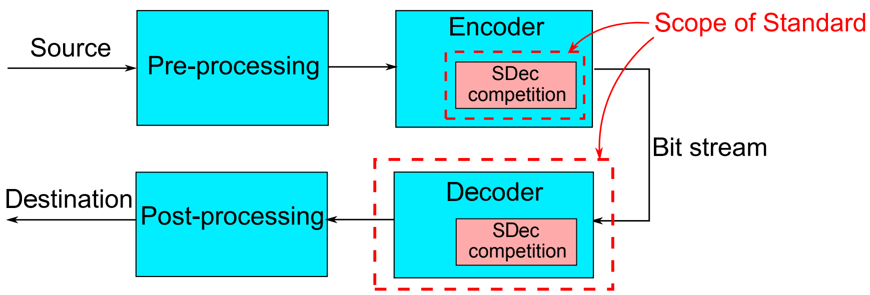


Figure 2.11 – *Partial standardization of the encoder to be compatible with the SDec scheme.*

More precisely, a standard for the SDec process related to the competition of coding modes must be defined to mandate the following points:

- The set of coding modes used in the SDec competition and the order in which they are competing against each other must be specified to ensure that the same coding mode that minimizes the R-D cost is computed at both encoder and decoder sides.
- The set of block candidates for the SDec reference must be specified so that an index transmitted by the encoder allows the decoder to determine the correct block candidate that is selected among the set.
- The evaluating criteria used in the SDec competition must also be specified. Indeed, the metric (e.g. R-D cost, SAD, SSE) used to compare the performance of different coding modes must be the same

at both encoder and decoder sides.

Indeed, this partial standardization of the encoder increases the number of requirements that video codecs must respect and can have an impact on the adoption of a new standard. Still, it can be justified and accepted in case a significant improvement in terms of compression performance is made thanks to the use of the SDec concept.

Conclusion

In this chapter, we have pointed out that reducing the signaling of coding modes and parameters are crucial for next generations of video codecs, since many more modes would be added to make the encoder more efficient and flexible, resulting in a significant increase in signaling overhead. Therefore, breaking with conventional approaches, an innovative coding scheme is proposed to reduce the signaling of competing coding modes and parameters by exploiting the processing ability of the decoder. Being calculated on the causal SDec reference PU rather than directly on the current PU, coding modes can be computed similarly in both encoder and decoder sides, saving thus their transmission.

This scheme allows consequently the integration of powerful tools that typically suffer from heavy signaling overhead. Its main drawback concerns the complexity as several coding modes must be tested on the reference during the SDec process. However, the scheme is entirely scalable, allowing the complexity to be adjusted depending on the available resources.

The selection of the SDec reference, which is a crucial step of the SDec scheme, is also discussed. Related consequences on the standardization of the SDec scheme are eventually presented.

In the next chapters, several practical applications of the general SDec scheme will be presented as a proof of concept.

3 SDec based coding scheme using Intra mode

In this chapter, we propose a practical application of the general SDec coding scheme described in the previous chapter to make use of some advantages of the SDec design. A practical application of the SDec scheme using only Intra coding mode is described, along with several studies concerning different parameters associated with SDec process. We implement in the HEVC test model software. Experimental results under JCT-VC common test conditions are also given.

3.1 Description

Derived from the general outline of the SDec scheme, the specific SDec scheme proposed in this chapter is simplified and has restriction concerning the number of coding modes in the SDec competition. Compared to the general scheme described in chapter 2, the number of available coding modes tested during the second step of SDec process is limited to HEVC Intra mode ($m = 1$). Parameters for Intra mode, which consist of 35 Intra directions, thus compete on each PU candidate ($q = 35$). We choose the Intra mode with its 35 directions for the following reasons:

- Syntax elements signaling Intra directions represent a significant proportion in the total bit stream, up to more than 20% for I-frames and 6% for P-, B-frames. Given the SDec principle which does not transmit the mode being used during SDec process, exploiting Intra directions as modes for the SDec scheme can have a large impact and allows to measure the potential of the whole approach.
- To improve the Intra coding efficiency of the texture, the number of competing Intra predictors are significantly increased through recent generations of video coding standard: only 1 predictor in MPEG-2, from 4 to 9 predictors in H.264/AVC, and up to 35 predictors in HEVC. This corresponds particularly to the motivation of using the SDec scheme previously mentioned.

Figure 3.1 illustrates in more detail the proposed specific SDec scheme using Intra coding mode. Up to 35 Intra directions are put in the R-D competition on the reconstructed SDec reference block P' which is re-encoded in Intra mode during the SDec competition. The direction minimizing the R-D cost on P' is eventually inherited to encode the current block P. Note that although the prediction is performed spatially, SDec mode in this specific scheme is to be assimilated to an Inter mode given that it exploits a reference block located in an already decoded frame.

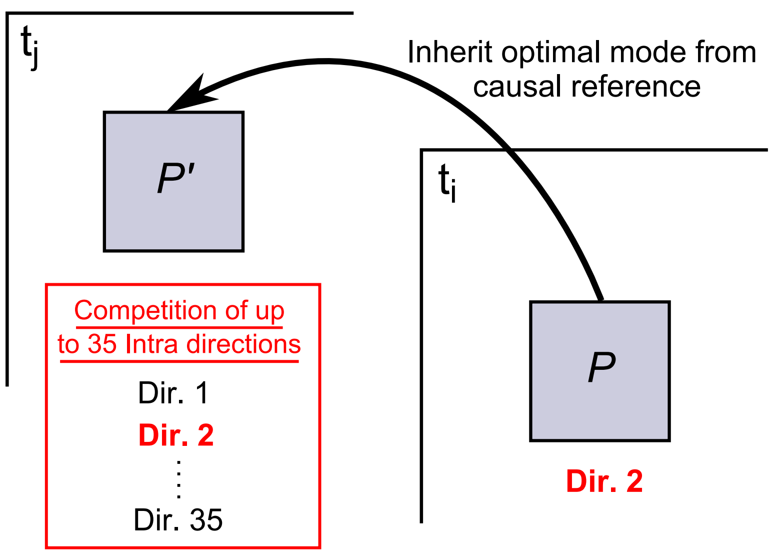


Figure 3.1 – *Principle of the SDec scheme using Intra mode: optimal Intra direction computed on P' in the already reconstructed frame at instant t_j is inherited to encode P in the current frame at instant t_i.*

Choosing Intra as the only mode for the SDec process, we propose that the syntax element *sdec_flag* signaling the SDec mode is inserted in the signaling scheme so that it will compete with the HEVC Intra mode, as represented in figure 3.2. Unlike HEVC Intra mode, SDec scheme using Intra coding mode does not require the signaling of the syntax elements *intra_dir_luma* and *intra_dir_chroma* to indicate the optimal Intra direction respectively for luminance and chrominance components, thus allowing bit rate savings.

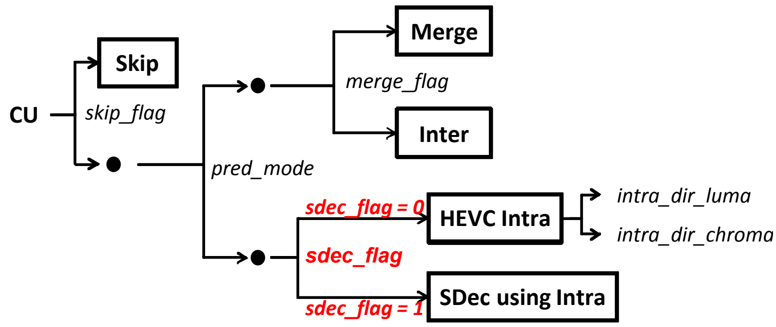


Figure 3.2 – *Signaling scheme for the introduced SDec syntax elements in the SDec scheme using Intra coding mode*

3.2 A study on different parameters of proposed scheme

In this section, we conduct several experiments to evaluate different intrinsic parameters of the SDec scheme. We first explain different CABAC based signaling methods for the new SDec syntax element *sdec_flag*. Then, we present experiments concerning the selection of the SDec reference block among several proposed block candidates. Next, we investigate the impact of the number of Intra directions used in the SDec competition on the compression performance. Finally, experiments aiming to evaluate the optimal number of candidates for the SDec reference are described. Compression performance of all tests are evaluated using Bjøntegaard Delta (B-D) rate (Bjøntegaard 2001) which represents the average difference between two R-D curves on the tested QP range.

Following test configuration is used for all the experiments:

Codec: HEVC, test model version 12.0 (HM12) QPs range: Medium bit rate QP = {22, 27, 32, 37} Test sequences: CTC test set for HEVC, shortened to 5 frames in All Intra Main (AI) and to 2 seconds in Random Access Main (RA) and Low Delay P Main (LP) Evaluation: Average B-D rate on all frames in AI and RA, and on P-frames in LP

3.2.1 Signaling method for the SDec flag

The introduced SDec mode is signaled using the syntax element *sdec_flag*. In this section, we present different experiments based on the

Context-Adaptive Binary Arithmetic Coding (CABAC) to improve the signaling efficiency of $sdec\_flag$.

Concerning the SDec reference, for the sake of simplification, we consider only a single candidate: the Colocated block found in the first frame of reference list L0 for P-, B-frames and the Left block of the current block (with the same size) for I-frames. The syntax element $sdec\_ref$ is thus not needed. Moreover, all 35 Intra directions are competing with each other during the SDec process.

We conducted tests where $sdec\_flag$ is signaled simply with one CABAC context or with three CABAC contexts defined based on the neighboring blocks as follows:

– None of Above and Left block is encoded with SDec mode,
– Only one block among Above and Left blocks is encoded with SDec mode,
– Both Above and Left blocks are encoded with SDec mode.

The comparison between the use of one or three CABAC contexts for signaling $sdec\_flag$ is given in table 3.1.

	AI	**RA**	**LP**
Class A	0.0	-0.1	0.0
Class B	0.0	0.0	0.0
Class C	0.0	0.0	0.0
Class D	0.0	0.0	0.0
Class E	0.0	-0.1	-0.1
Class F	-0.1	-0.1	-0.1
Overall	**0.0**	**0.0**	**0.0**
Max	-0.2	-0.1	-0.3
EncTime	99%	99%	100%
DecTime	102%	100%	100%

Table 3.1 – *Gain when signaling sdec_flag with 3 CABAC contexts, compared to the use of 1 CABAC context (Ref: SDec scheme with sdec_flag signaled with 1 context)*

According to table 3.1, using three CABAC contexts provides a slight gain increase for some test sequence classes compared with the use of only one context. Note that the initial probability of all contexts are set arbitrarily to 50%. We also conducted tests to set a suitable value as the initial probability for each CABAC context: the probability of $sdec\_flag$ being equal to 1 is plotted for each context, linear regression is then used to compute the *slope* and the *offset* which are parameters to calculate CABAC initial values. However, there is no improvement in average compared to the use of the arbitrary value of 50%. Indeed, due to the large number of blocks in each frame for most sequences in the test set, the probability models for CABAC contexts converge regardless of the initial value.

Table 3.2 presents in more detail the gain performance of the SDec scheme when $sdec\_flag$ is signaled with three CABAC contexts, compared to the reference HEVC. Different coding rates are used to evaluate the performance of the proposed method: low bit rate (LBR) QP={27,32,37,42}, medium bit rate (MBR) QP={22,27,32,37}, high bit rate

(HBR) QP={17,22,27,32}.

	AI			RA			LP		
	LBR	MBR	HBR	LBR	MBR	HBR	LBR	MBR	HBR
Class A	-0.1	-0.1	-0.1	-0.3	-0.2	-0.2	-1.2	-0.7	-0.4
Class B	-0.1	-0.2	-0.1	-0.4	-0.3	-0.2	-1.1	-0.7	-0.4
Class C	-0.2	-0.2	-0.2	-0.3	-0.2	-0.2	-0.9	-0.5	-0.2
Class D	-0.1	-0.1	0.0	-0.1	-0.1	-0.1	-0.8	-0.4	-0.2
Class E	-0.2	-0.2	-0.1	-0.2	-0.2	-0.2	-1.1	-0.7	-0.5
Class F	1.2	-0.1	-0.4	-0.3	-0.3	-0.4	-1.9	-1.6	-1.0
Overall	**0.1**	**-0.1**	**-0.2**	**-0.3**	**-0.2**	**-0.2**	**-1.2**	**-0.8**	**-0.5**
Max	-0.5	-0.5	-0.8	-1.0	-0.8	-1.1	-5.2	-3.2	-2.0
EncTime	460%			317%			219%		
DecTime	486%			321%			306%		

Table 3.2 – *SDec compression performance when signaling sdec_flag with 3 CABAC contexts based on neighboring blocks (Ref: HM12)*

We remark that the SDec scheme using Intra mode is often most efficient at low bit rate, achieving for example under LP configuration -1.2% in average gain in LBR compared with -0.8% in MBR. There is an exception for the sequence "*SlideShow*_1280 × 720" of the class F in AI configuration, which yields a significant gain of -0.8% in HBR and a severe loss of 5.5% in LBR. This is due to the particular visual content of the sequence, having very light and repetitive pattern that is easily lost by degradation in LBR, penalizing thus the SDec scheme which inherits the mode of neighboring blocks in AI configuration.

Generally, the definition of contexts is preferred to be independent from previously encoded blocks. Therefore, we propose another method to define CABAC contexts for the syntax element *sdec_flag*. Instead of using the coding mode of neighboring blocks, the size of the current block is exploited in order to remove the dependency. There are four CABAC contexts corresponding to four block sizes:

- Size 64 × 64,
- Size 32 × 32,
- Size 16 × 16,
- Size 8 × 8.

The performance of the proposed method using four size-depending CABAC contexts for *sdec_flag* is given in table 3.3 while taking the version using three CABAC contexts based on neighboring blocks as the reference.

According to table 3.3, the use of four CABAC contexts based on the size of the current block provides some improvements in coding gain, especially for class F in LP configuration. However, we observe loss for several test sequence classes in RA configuration. The decoding time is also slightly increased. Therefore, we will keep the signaling of *sdec_flag* by using three CABAC contexts for the next experiments.

	AI			RA			LP		
	LBR	**MBR**	**HBR**	**LBR**	**MBR**	**HBR**	**LBR**	**MBR**	**HBR**
Class A	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0
Class B	0.0	0.0	0.0	0.0	0.0	-0.1	0.0	0.1	0.1
Class C	0.0	0.0	0.0	0.1	0.1	0.0	0.0	0.0	-0.1
Class D	0.0	0.0	0.0	0.0	0.0	-0.1	0.2	0.1	0.0
Class E	-0.2	-0.1	0.0	0.1	0.1	0.1	0.0	0.1	0.1
Class F	0.1	0.0	-0.1	0.1	0.1	0.0	-0.4	-0.4	-0.6
Overall	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**-0.1**
EncTime	100%			100%			100%		
DecTime	108%			104%			103%		

Table 3.3 – *SDec compression performance when signaling sdec_flag using 4 CABAC contexts based on current block sizes (Ref: 3 CABAC contexts based on neighboring blocks)*

3.2.2 Block candidates for the SDec reference

For simplification purpose, we consider only a single candidate block for the SDec reference in this section. Therefore there is no need to use the syntax element *sdec_ref* that indicates the index of selected candidate among several ones. Different blocks are proposed to find the one that yields the highest gain. All 35 HEVC Intra directions are used in the SDec competition.

Proposed block candidate to be the SDec reference are classified into following categories:

- Spatial pre-identified neighboring blocks:

Blocks surrounding the current block, located in the causal region of the current frame and having the same size as the current block: ***Left* (L)**, ***Above* (A)**, ***AboveLeft* (AL)**, ***BottomLeft* (LB)** and ***AboveRight* (AR)** as shown in figure 3.3. Note that not all of those blocks are already encoded when the current block is processed. If a block candidate is not available to be the SDec reference, the SDec scheme is skipped.

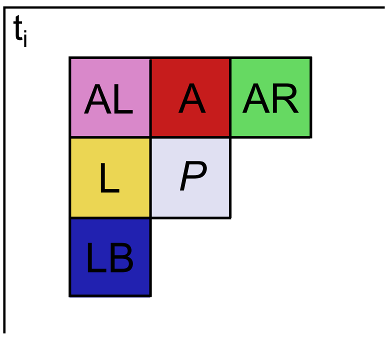


Figure 3.3 – *Proposed pre-identified spatial neighboring blocks as SDec reference candidates for the current block P in the frame at instant t_i.*

- Temporal pre-identified blocks:

For Inter frames (P- or B-frames), there is the possibility to choose a temporal block which is found in a previously decoded frame to be the SDec reference. Different pre-identified candidates are proposed as follows and are depicted in figure 3.4:

- *Col*: the colocated block in the first frame of the reference picture list L0.
- *ColL, ColA, ColAL, ColLB, ColAR*: different neighboring blocks of *Col*.

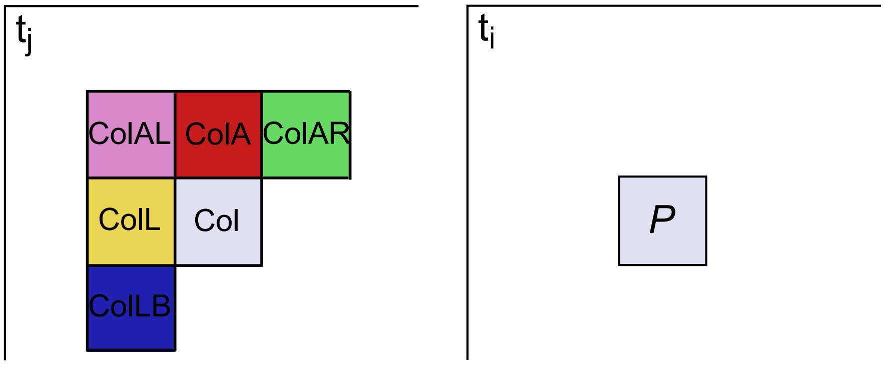


Figure 3.4 – *Proposed pre-identified temporal blocks in a reconstructed frame at instant t_j as the SDec reference candidates for the current block P in the frame at instant t_i.*

- Temporal blocks identified by Merge motion vector (MV) candidates:

Aside from using pre-identified blocks, another way to select a block candidate for the SDec reference is to exploit Merge MV candidates, which point to a position in a previously reconstructed frame. Following candidates are proposed:

- *BlkMrgMvL, BlkMrgMvA, BlkMrgMvAL, BlkMrgMvLB, BlkMrgMvAR, BlkMrgMvColRB, BlkMrgMvCol*: different blocks pointed by the Merge MV candidate respectively of the block Left, Above, AboveLeft, LeftBottom, AboveRight, RightBottom, ColocatedRightBottom and Colocated.
- *BlkMrgMvMedian*: the block derived from exploiting all Merge MV candidates and determined by following steps:
  - Establish a modified Merge list removing both the restriction on the maximum number of candidates (5) and the filling process that generates combined candidates. Duplicative MVs are allowed.
  - X and Y coordinates of all Merge MV candidates in the list are sorted independently in the ascending order. The median values among X and Y sorted lists are selected as components of the final MV that points to the block.

- Temporal block identified by Template Matching technique:

We also propose to exploit block computed by the template matching technique to be the SDec reference. The causal surrounding region located in the Left and Top sides of the current block is used as a search template. Figure 3.5 illustrates the template for a current block with size n. Parameter e represents the thickness of the template.

Using this template shape, we search in the previously reconstructed frame for the block that has its template most similar to the template of the current block in terms of SSE metric. Instead of searching in the entire frame which requires a very high runtime, we restrict to a small region with a configurable radius r around the current block. Inside that search zone, we perform the template matching search in a spiral order as shown

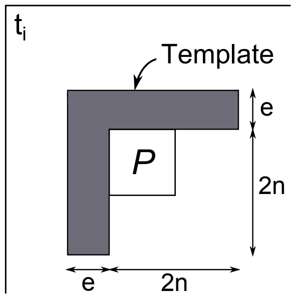


Figure 3.5 – *The neighbor surrounding region of the $n \times n$ current block located in the frame at instant t_i is used as template with thickness e.*

in figure 3.6. This exhaustive search is stopped prematurely when a zero SSE value is obtained, i.e. when an identical template is found. This spiral search can help reducing the complexity for sequences containing slow or no motion since the position of an object does not vary significantly from the previously reconstructed frame to the current frame.

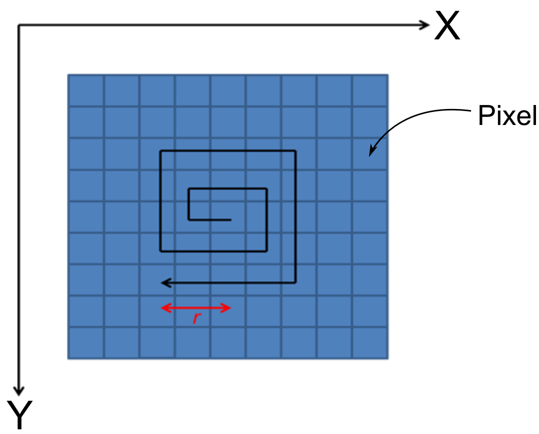


Figure 3.6 – *Spiral template matching search order from pixel to pixel within search region of radius r.*

In the end of the template matching process, the block that minimizes the template distortion is then used as a candidate for the SDec reference. We denote this block by ***BlkTplMtch***. The template thickness e and search radius r are both set to 8 pixels since this configuration yields the best compromise between the coding gain and the complexity according to our test on different sets of (e, r).

Table 3.4 summarizes the performance of the SDec scheme using Intra mode with different cases where each of the previously proposed blocks is used as the unique candidate for the SDec reference. The average B-D rate gain is given compared with the reference HM12.

We observe that for the AI configuration, A block yields the best performance among proposed spatial neighboring candidates, with an average bit rate saving of -0.2%. In configurations RA and LP, Col block is the best candidate among all proposed temporal candidates, yielding an average B-D gain of -0.2% and -0.8% respectively. By analyzing the visual content of tested sequences, it is observed that using Col block as the candidate for the SDec reference is suited for most of the cases, including sequences

	Candidate block	AI	RA	LP
Spatial pre-identified blocks	L	-0.1	-0.1	-0.3
	A	-0.2	-0.2	-0.4
	AL	-0.1	-0.1	-0.3
	AR	-0.1	-0.1	-0.2
	LB	0.0	0.0	-0.1
Temporal pre-identified blocks	Col		-0.2	-0.8
	ColL		-0.1	-0.4
	ColA		-0.1	-0.4
	ColAL		-0.1	-0.3
	ColAR		-0.1	-0.2
	ColLB		0.0	-0.1
Temporal blocks identified by Merge MVs	BlkMrgMvL		-0.1	-0.4
	BlkMrgMvA		-0.1	-0.4
	BlkMrgMvAL		-0.1	-0.4
	BlkMrgMvAR		-0.1	-0.3
	BlkMrgMvLB		0.0	-0.1
	BlkMrgMvColRB		-0.1	-0.5
	BlkMrgMvCol		0.0	-0.7
	BlkMrgMvMedian		-0.1	-0.6
Temporal blk. identified by template matching	BlkTplMtch		-0.1	-0.6

Table 3.4 – *Compression performance of SDec scheme using Intra mode with different blocks used as unique candidate for SDec reference (Ref: HM12)*

with little movement (e.g. sequence "Johnny_720p" with videoconferencing content) or movement that is difficult to predict by conventional temporal coding modes (e.g. sequence "Cactus_1080p" containing rotation). For sequences with easily predictable motion (e.g. sequence "SteamLocomotiveTrain_1600p" depicting a train moving forward), block candidates identified using Merge MVs (e.g. BlkMrgMvCol, BlkMrgMvMedian) have a better performance.

We also remark that using block computed by template matching technique gives a fairly good result (-0.6% in LP). However, the runtime is excessively long due to the template search process. Therefore, we will exploit other blocks candidates in experiments evaluating the use of multiple candidates for the SDec reference that will be mentioned in next sections.

3.2.3 Number of Intra directions used in the SDec competition

We propose to study the impact of the number of Intra directions on the coding performance of the proposed SDec scheme. By gradually reducing that number, instead of using all 35 available directions, we perform several tests where 19, 7 and 3 directions are used. Planar and DC are always retained since they are most often selected. Among 33 angular Intra directions indexed from 2 to 34, depending on the restrained number of directions, retained directions are equally distributed in space so that there is no bias in any direction. More precisely, the Intra directions used for each test are given by their indices as follows:

- 35 directions: All 35 available Intra directions are exploited.
- 19 directions: Planar (0), DC (1), 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34.

– 7 directions: Planar (0), DC (1), 2, 10, 18, 26, 34.
– 3 directions: Planar (0), DC (1) and Vertical (26).

We still use only one candidate to be the SDec reference for the sake of simplification. From the previous result concerning the selection of suitable candidate, we choose the Colocated block and the Above block respectively for P-, B- frames and I-frames as optimal candidates to be the SDec reference. The average coding gain for each of four tested cases is given in table 3.5.

	3 directions			7 directions			19 directions			35 directions		
	AI	**RA**	**LP**	**AI**	**RA**	**LP**	**AI**	**RA**	**LP**	**AI**	**RA**	**LP**
Class A	-0.2	-0.1	-0.4	-0.2	-0.2	-0.4	-0.2	-0.2	-0.7	-0.1	-0.2	-0.7
Class B	-0.2	-0.2	-0.3	-0.2	-0.2	-0.5	-0.2	-0.2	-0.6	-0.2	-0.3	-0.7
Class C	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.4	-0.3	-0.2	-0.5
Class D	-0.1	-0.2	-0.2	-0.1	-0.2	-0.2	-0.1	-0.1	-0.3	-0.1	-0.1	-0.4
Class E	-0.2	-0.2	-0.3	-0.2	-0.1	-0.2	-0.2	-0.3	-0.7	-0.4	-0.2	-0.7
Class F	0.2	-0.1	-0.1	-0.1	-0.2	-1.7	-0.1	-0.2	-1.8	-0.2	-0.3	-1.6
Overall	**-0.1**	**-0.2**	**-0.3**	**-0.2**	**-0.2**	**-0.6**	**-0.1**	**-0.2**	**-0.8**	**-0.2**	**-0.2**	**-0.8**
Max	-0.4	-0.6	-0.6	-0.5	-0.6	-3.3	-0.4	-0.7	-3.6	-0.6	-0.8	-3.2
Min	0.7	0.1	0.1	0.4	0.1	0.1	0.3	0.2	0.0	0.0	0.3	-0.2
EncTime	167%	143%	129%	221 %	173 %	148 %	338%	244 %	184%	458%	317%	219%
DecTime	170%	144%	141%	230 %	178 %	152 %	352%	226 %	223%	517%	321%	306%

Table 3.5 – *SDec compression performance when using different number of Intra directions in SDec competition (Ref: HM12).*

We observe that increasing the number of Intra directions in the SDec competition generally increases the gain in average. The tested case using 35 directions gives best results, yielding -0.2%, -0.2% and -0.8% average bit rate savings respectively in AI, RA and LP configurations. However, the runtime for both encoder and decoder sides is also significantly longer due to the exhaustive check of all 35 Intra directions on the SDec reference during the SDec process.

In order to reduce the complexity, we propose to restrain the number of Intra directions while adaptively selecting suitable directions for the SDec competition. Inspired from the HM software, we create a pre-selection phase with the objective to evaluate only a limited number n of Intra directions that minimize block distortion among 35 available directions. More precisely, all 35 directions are firstly tested on the SDec reference using a simplified distortion evaluation (Hadamard transform) and without performing the inverse tranform-quantization process. Only n best directions are retained for the real encoding of the SDec reference.

We conduct tests with number n of Intra directions retained after the pre-selection phase equal to 3 or 8. The average coding gain is given in table 3.6.

We observe that the proposed method with number n of Intra directions equal to 8 provides an acceptable compromise between the compression performance and the runtime, giving a good coding gain without suffering too much from the excessive runtime.

	3 among 35			8 among 35		
	AI	**RA**	**LP**	**AI**	**RA**	**LP**
Class A	0.1	-0.1	-0.7	-0.1	-0.2	-0.8
Class B	0.1	-0.2	-0.6	-0.1	-0.3	-0.7
Class C	0.2	0.0	-0.3	-0.2	-0.2	-0.5
Class D	0.5	0.1	-0.2	-0.1	-0.2	-0.4
Class E	0.1	0.2	-0.2	-0.2	-0.2	-0.8
Class F	0.9	0.6	-1.1	-0.4	-0.3	-1.4
Overall	**0.3**	**0.1**	**-0.5**	**-0.2**	**-0.2**	**-0.7**
Max	-0.2	-0.7	-3.4	-0.8	-0.7	-3.20
Min	1.5	1.2	0.2	0.0	0.0	-0.2
EncTime	234%	196%	158%	271%	208%	162%
DecTime	257%	188%	188%	253%	203%	196%

Table 3.6 – *Approach using fast adaptive selection for restrained number of Intra directions in SDec competition (Ref: HM12).*

3.2.4 Number of candidates for the SDec reference

Instead of using a single candidate for the SDec reference that gives only one Intra direction to be inherited on the current block, adding more candidates provides additional Intra directions, increasing thus the probability that the inherited direction is also optimal for the current block. Potential further gain can therefore be obtained.

Figure 3.7 shows the SDec signaling scheme when there are more than one candidate for the SDec reference. Comparing to figure 3.2, an additional syntax element *sdec_ref* is signaled for every block encoded in SDec mode to indicate which candidate is selected to be the SDec reference.

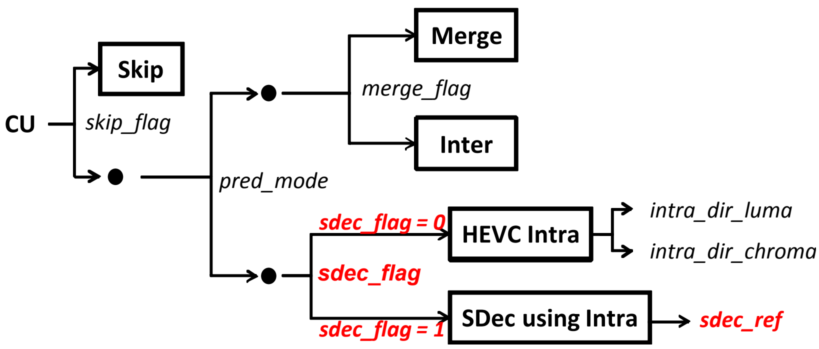


Figure 3.7 – *Signaling scheme for sdec_ref syntax element in case of multiple candidates for the SDec reference.*

This index of selected candidate *sdec_ref* must be transmitted in the bit stream so that the decoder can determine the correct SDec reference on which the encoder conducted the competition of all SDec coding modes and parameters. The higher number of candidates, the more significant signaling overhead is required for transmitting *sdec_ref*.

In the following tests where different number of block candidates are evaluated, the number of Intra directions is set to 8 using the adaptive pre-selection phase mentioned in the previous test.

2 candidates:

Since there are two block candidates, for a block that is encoded in SDec mode, we signal *sdec_ref* with 1 bit using a CABAC context model as shown in figure 3.8. The initial value of the context is set arbitrarily to 50%.

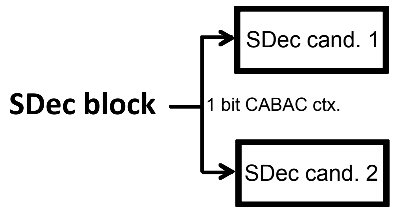


Figure 3.8 – *Signaling of sdec_ref in case of 2 candidates for the SDec reference.*

If both block candidates yield the same optimal Intra direction, it becomes equivalent with the use of a single candidate. Therefore, *sdec_ref* is not signaled in this case. At the decoder side, it still remains perfectly decodable: the decision whether or not to parse *sdec_ref* can be indeed correctly obtained by implementing the SDec module in the parsing phase where block information is extracted from the bit stream. Furthermore, from the list of proposed block candidates, if we consider each block as the second candidate until it yields a different Intra direction regarding the first candidate, it is observed that there is no improvement in average while the runtime is increased due to the SDec competition on additional candidates.

For configuration AI, table 3.7 presents the SDec performance when using two candidates for the SDec reference. The first candidate is the Above block since it provides the best gain during previous experiments with a unique candidate. Other spatial neighboring blocks (L, AL, AR, LB) are added as the second candidate. The percentage distribution of each candidate is also given.

		Gain	Max	Min	%Cand.1	%Cand.2
A		**-0.2**	-0.6	0.2		
A &	L	**-0.5**	-1.5	-0.1	52.1%	47.9%
	AL	**-0.4**	-1.1	0.0	67.9%	32.1%
	AR	**-0.4**	-1.0	-0.1	72.5%	27.5%
	LB	**-0.3**	-0.9	-0.1	85.2%	14.8%

Table 3.7 – *SDec compression performance when using 2 candidates for the SDec reference, configuration AI.*

According to table 3.7, compared with the use of block Above as the unique candidate (-0.2%), adding a second candidate improves the coding gain in overall, with an average gain of at least -0.3% for combining with block LeftBottom and up to -0.5% with the additional use of block Left. The maximum gain is also increased. This gain improvement proves that adding a second block candidate increases the likelihood that information inherited from the SDec reference is relevant to the current block. Being the best second candidate, the Left block provides more useful information to be exploited compared to other blocks, with a distribution percentage of 47.9% when combined with the Above block.

For configurations RA and LP, we use Colocated block as the first candidate for the SDec reference. Other proposed spatial and temporal blocks are added as the second candidate. Coding gains are given in table 3.8.

			Gain	Max	Min	%Cand.1	%Cand.2
Col		RA	**-0.2**	-0.7	0.0		
		LP	**-0.7**	-3.2	-0.2		
Col &	L	RA	**-0.3**	-1.1	0.1	57.8%	42.2%
		LP	**-0.9**	-4.7	-0.1	78.5%	21.5%
	A	RA	**-0.4**	-1.0	0.0	56.4%	43.6%
		LP	**-0.9**	-4.1	-0.3	77.5%	22.5%
	AL	RA	**-0.3**	-0.9	0.1	66.3%	33.7%
		LP	**-0.9**	-5.0	-0.1	83.2%	16.8%
	AR	RA	**-0.3**	-0.9	0.1	77.0%	23.0%
		LP	**-0.9**	-4.1	-0.2	88.7%	11.3%
	LB	RA	**-0.2**	-0.8	0.3	88.7%	11.3%
		LP	**-0.9**	-5.0	-0.2	94.4%	5.6%
	ColL	RA	**-0.3**	-0.9	0.1	69.0%	31.0%
		LP	**-0.9**	-3.8	-0.2	78.7%	21.3%
	ColA	RA	**-0.2**	-0.7	0.0	69.3%	30.7%
		LP	**-0.9**	-3.9	-0.1	80.8%	19.2%
	ColAL	RA	**-0.3**	-0.9	0.2	71.5%	28.5%
		LP	**-0.8**	-3.8	-0.2	84.3%	15.7%
	ColAR	RA	**-0.3**	-0.8	0.0	79.7%	20.3%
		LP	**-0.9**	-3.9	-0.1	88.9%	11.1%
	ColLB	RA	**-0.2**	-0.7	0.3	90.4%	9.6 %
		LP	**-0.9**	-4.1	-0.1	94.4%	5.6%
	BlkMrgMvL	RA	**-0.3**	-0.8	0.0	77.5%	22.5 %
		LP	**-0.9**	-4.4	-0.1	76.0%	24.0%
	BlkMrgMvA	RA	**-0.2**	-0.8	0.1	77.3%	22.7 %
		LP	**-0.9**	-4.1	-0.2	75.5%	24.5%
	BlkMrgMvAL	RA	**-0.3**	-0.9	0.0	76.0%	24.0 %
		LP	**-0.9**	-4.0	-0.1	75.0%	25.0%
	BlkMrgMvAR	RA	**-0.3**	-0.8	0.1	84.0%	16.0 %
		LP	**-0.9**	-4.3	-0.1	83.2%	16.8%
	BlkMrgMvLB	RA	**-0.3**	-0.8	0.0	92.8%	7.2 %
		LP	**-0.9**	-4.1	0.0	92.0%	8.0%
	BlkMrgMvColRB	RA	**-0.3**	-0.8	0.1	85.4%	14.6 %
		LP	**-0.9**	-4.0	-0.1	72.6%	27.4%
	BlkMrgMvCol	RA	**-0.3**	-0.8	0.0	81.3%	18.7 %
		LP	**-1.0**	-4.2	-0.2	65.2%	34.8%
	BlkMrgMvMedian	RA	**-0.3**	-0.9	0.2	69.2%	30.8 %
		LP	**-1.0**	-4.0	-0.1	60.9%	39.1 %

Table 3.8 – *SDec compression performance when using 2 candidates for the SDec reference, configurations RA and LP.*

The results show that, compared with the use of the Colocated block as the unique candidate for the SDec reference, adding a second candidate systematically improves the coding gain for configurations RA and LP. The maximum gain is increased as well. This improvement in performance is due to the improvement in coding quality with the use of the second candidate and also to the efficient signaling of introduced syntax element $sdec\_ref$ which is signaled with as least bit as possible (using only one bit, or even not signaled at all when both candidates yield the same Intra direction). Signaling *sdec_ref* costs respectively 0.2%, 0.1% and 0.1% of the total bit stream for configurations AI, RA and LP. Finally, two following block combinations are observed to yield highest bit rate savings in average:

– Col combined with BlkMrgMvMedian or BlkMrgMvCol, providing gains of -0.3% and -1.0% respectively in RA and LP.
– Col combined with A, providing gains of -0.4% and -0.9% respectively in RA and LP.

3 candidates:

After obtaining the best combination of two candidates for the SDec reference, we increase again the number of candidates to three. The signaling scheme of *sdec_ref* for index among three candidates is illustrated in figure 3.9. Each bit indicating the index of the selected candidate is signaled using a CABAC context model with initial value set to 50%.

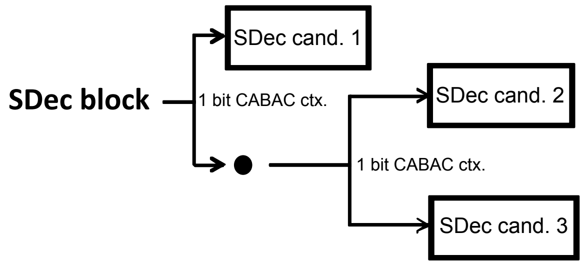


Figure 3.9 – *Signaling of sdec_ref in case of 3 candidates for the SDec reference.*

If different candidates yield the same optimal Intra direction, there are fewer indexes for *sdec_ref* to be signaled. In this case, we re-use the signaling scheme previously proposed for one (index not signaled) or two candidates (using only one bit). For signaling *sdec_ref*, we choose the scheme with hierarchical bits in order to privilege the first candidate from the second and third candidates. Indeed, it is observed that the first candidate possesses a significant percentage distribution according to previous test with two candidates for the SDec reference.

For configuration AI, from the best combination of two candidates (Above combined with Left blocks) according to the previous test, different blocks (AL, AR) are added and tested as the third candidate to be the SDec reference. The results are summarized in table 3.9.

		Gain	Max	Min	%Cand.1	%Cand.2	%Cand.3
A & L &	AL	**-0.6**	-2.2	-0.1	42.5%	40.6%	16.9 %
	AR	**-0.5**	-1.1	-0.1	45.0%	39.1%	15.9%

Table 3.9 – *SDec compression performance when using 3 candidates for the SDec reference, configuration AI.*

Experimental results show that adding AboveLeft block as the third block candidate for the SDec reference improves further the coding performance, with average gain of -0.6% compared with -0.5% when using only two candidates Above and Left blocks. Figure 3.10 illustrates a frame from the sequence "ChinaSpeed_1024×768". Blocks encoded in SDec mode are highlighted in different colors depending on the index of candidate selected as SDec reference: yellow, blue and red respectively for the first, second and third candidates. By observing the visual distribution for each candidate, we can remark a fairly good number of blocks encoded in SDec mode using additional third candidate AboveLeft. For the case of using AboveRight block as the third candidate, no further increase in average gain is obtained.

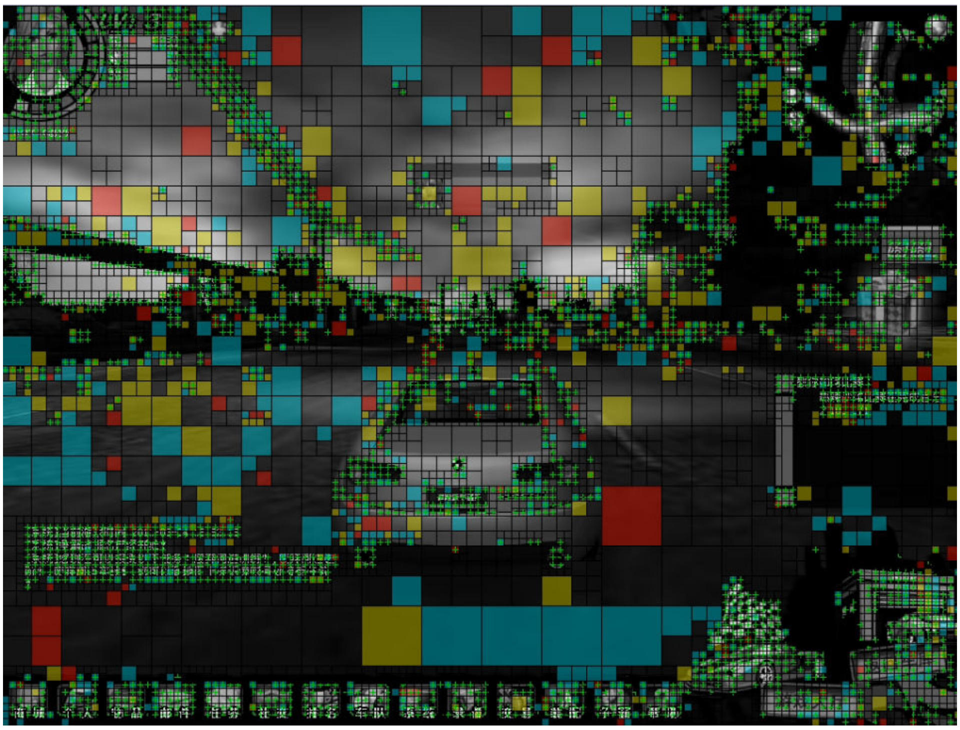


Figure 3.10 – *Sequence "ChinaSpeed_1024×768" with blocks encoded in SDec mode highlighted in yellow, blue and red for exploiting respectively the first, the second and the third block candidate to be the SDec reference - AI configuration.*

For configurations RA and LP, we propose to add the third candidate (L, A, ColL, ColA) to the combination of Col and BlkMrgMvMedian blocks. The results are given in table 3.10.

			Gain	Max	Min	%Cand.1	%Cand.2	% Cand.3
Col & BlkMrgMvMedian &	L	RA	**-0.3**	-1.2	0.0	45.8%	33.5%	20.7%
		LP	**-0.9**	-4.5	0.0	54.8%	31.1%	14.2 %
	A	RA	**-0.4**	-1.1	0.1	52.9 %	23.8%	23.3%
		LP	**-1.0**	-3.5	-0.1	55.9%	31.6%	12.5%
	ColL	RA	**-0.3**	-1.0	0.0	52.9%	23.9%	23.2 %
		LP	**-0.9**	-3.9	-0.1	55.7%	30.9 %	13.5%
	ColA	RA	**-0.3**	-1.0	0.1	45.0%	34.9%	20.1%
		LP	**-1.0**	-4.7	-0.2	53.8%	31.1%	15.1 %

Table 3.10 – *SDec compression performance when using 3 candidates for SDec reference, configurations RA and LP.*

We observe that adding a third candidate under configurations RA and LP does not improve further the gain in average, providing at best gains of -0.4% and -1.0% respectively for those two configurations, similar to the use of two candidates. This can be explained by the increase in signaling cost of *sdec_ref*, which is respectively 0.3%, 0.1% and 0.2% of total bit stream for configurations AI, RA and LP.

4 candidates:
For signaling the index *sdec_ref* among four block candidates for the SDec reference, we use the signaling scheme illustrated in figure 3.11. One CABAC context model with initial value of 50% is used to signal each bit.

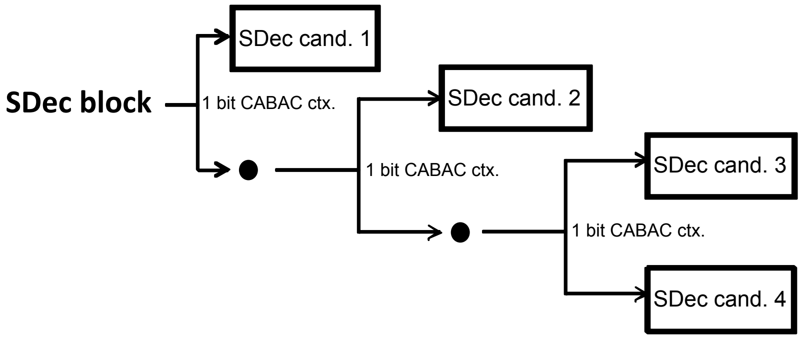


Figure 3.11 – *Signaling of sdec_ref in case of 4 candidates for the SDec reference*

If the same optimal Intra direction is yielded by different candidates, less signaling overhead is required to signal the index *sdec_ref*. The signaling scheme of n candidates ($n < 4$) mentioned previously is then exploited. Similar to the previous case of three candidates, signaling scheme with hierarchical bits is used to signal syntax element *sdec_ref*. We can besides confirm the efficiency of the chosen signaling scheme for *sdec_ref* since the usage percentage for each candidate is not equally distributed according to the experimental results concerning two and three candidates.

Table 3.11 provides the results when four candidates are used for the SDec reference.

	A & L & AL & AR	Col & BlkMrgMvMedian & A & ColA	
	AI	RA	LP
Gain	**-0.6**	**-0.4**	**-0.9**
Max	-1.3	-1.1	-4.2
Min	-0.1	0.0	-0.2
EncTime	768%	337 %	346 %
DecTime	1763%	408 %	638 %

Table 3.11 – *SDec compression performance when using 4 candidates for the SDec reference.*

Compared to the use of three candidates, it is observed that the average gain is not further increased under all tested configurations. The exploitation of an additional fourth candidate for the SDec reference does not provide any benefice because the signaling cost of syntax element *sdec_ref* becomes too significant, reaching respectively 0.4%, 0.2% and 0.2% of total bit rate for configurations AI, RA and LP.

Summary:

We summarize in table 3.12 the results concerning tests on different number of candidates for the SDec reference. The number of Intra directions used in the SDec competition is set to 8 with the pre-selection phase as described in 3.2.3. Optimal combinations of blocks candidates for each case are given as follows:

- 1 candidate:
  - I-frames: A
  - P-, B-frames: Col
- 2 candidates:
  - I-frames: A and L
  - P-, B-frames: Col and A, Col and BlkMrgMvMedian, Col and BlkMrgMvCol.
- 3 candidates:
  - I-frames: A, L and AL
  - P-, B-frames: Col, A and BlkMrgMvMedian
- 4 candidates:
  - I-frames: A, L, AL and AR
  - P-, B-frames: Col, A, BlkMrgMvMedian and ColA

	1 cand.			2 cands.			3 cands.			4 cands.		
	AI	RA	LP	AI	RA	LP	AI	RA	LP	AI	RA	LP
Gain	**-0.2**	**-0.2**	**-0.7**	**-0.5**	**-0.4**	**-0.9**	**-0.6**	**-0.5**	**-1.0**	**-0.6**	**-0.4**	**-0.9**
Max	-0.8	-0.7	-3.2	-1.5	-1.0	-3.7	-2.2	-1.1	-4.2	-1.3	-1.1	-4.2
Min	0.0	0.0	-0.2	-0.1	0.1	0.0	-0.1	0.0	-0.2	-0.1	0.0	-0.2
EncTime	271%	208%	162%	459%	225%	227%	638%	281%	289%	768 %	337 %	436 %
DecTime	253%	203%	196 %	858%	286%	346%	1356 %	407 %	548 %	1763 %	408 %	638 %

Table 3.12 – *Summary of the SDec performance when using different number of candidates for the SDec reference (Ref: HM12).*

In conclusion, experimental results show that increasing the number of competing candidates for the SDec reference (up to three) indeed provides additional useful information to be exploited, resulting in a coding gain improvement. From four candidates, the signaling cost of index $sdec\_ref$ for selected candidate becomes too high to be compensated by the accuracy in texture prediction, penalizing in turn the compression performance.

3.2.5 Selection of the SDec reference based on the Intra most probable modes

Intra most probable modes (MPM) is an efficient technique to signal the direction for a block encoded in HEVC Intra mode. Indeed, statistics show that the probability that MPM mechanism can correctly predict the Intra direction is as high as 70%. Furthermore, when MPM is used, very few bits are transmitted in the bit stream to signal the Intra direction. This observation suggests us to perform tests where block candidates for the SDec reference are only retained if they provide an Intra direction different to all three MPM values. Otherwise, if an Intra direction is among MPM values, the HEVC Intra coding mode is efficient and could outperform the SDec mode.

The number of candidates for the SDec reference is limited to two in maximum in order to avoid the costly signaling of the syntax element *sdec_ref*. From the list of four best block candidates obtained in the previous section, a block candidate is retained only if its optimal Intra direction is not included in the three MPM values. There is an exception for P- and B-frames where the first candidate is always the Colocated block due to its good performance. The second candidate is then chosen among the remaining blocks.

As shown in 3.13, the proposed method increases the maximum gain when tested on the HEVC test set, compared with the best combination of two candidates {Col & ColAbove} mentioned in the previous section. However, no further improvement in average gain is obtained. Furthermore, the runtime is increased because the SDec competition process is conducted on additional blocks until a block candidate delivers an optimal Intra direction that is not among the MPM values.

Sequences class	**LP**
Class A	0.0
Class B	0.1
Class C	0.0
Class D	0.2
Class E	0.0
Class F	-0.5
Average	**0.0**
Max gain	-1.5
EncTime	115%
DecTime	115%

Table 3.13 – *Block candidate is only selected to be the SDec reference if its optimal Intra direction is not among Intra MPM values - 2 candidates for the SDec reference (Ref: normal SDec scheme with {Col & ColAbove} as block candidates).*

With the same idea of exploiting the MPM values, we also propose to reduce the complexity by creating a shortcut at the encoder side. The SDec mode is only activated when the optimal Intra direction inherited from the SDec reference is not among all MPM values. Indeed, in the other case where the optimal Intra direction is among the MPM values, the HEVC Intra mode would be efficient thanks to the MPM mechanism and thus, it is possible to deactivate the SDec mode to reduce the complexity without suffering significant loss.

We conduct a test using only a single candidate for the SDec reference: Colocated block for P-, B-frames and Above block for I-frames. If the optimal Intra direction computed on the SDec reference is included among three MPM values, the SDec process of encoding the current block is stopped. Table 3.14 gives the experimental result of the shortcut approach when compared with the normal SDec scheme as reference. We can observe that there is indeed a decrease in runtime at both the encoder and the decoder sides due to the smaller number of blocks encoded in the SDec mode. Unfortunately, the coding gain is also reduced quite significantly under the configuration AI.

Sequences class	AI	RA	LP
Class A	0.1	0.1	0.3
Class B	0.2	0.1	0.3
Class C	0.2	0.1	0.1
Class D	0.2	0.1	0.2
Class E	0.4	0.3	0.1
Average	**0.2**	**0.1**	**0.2**
Max gain	0.0	-0.1	-0.1
EncTime	92%	95%	95%
DecTime	55%	72%	74%

Table 3.14 – *Shorcut approach deactivating the SDec mode when the optimal Intra direction is among Intra MPM values, a single candidate for the SDec reference is used (Ref: SDec scheme without the shortcut).*

3.3 Best configuration

According to the previous experiments, using more candidates increases the likelihood that the inherited Intra direction, which is optimal for SDec reference P', is also efficient to encode the current PU P. However, more signaling overhead is required to transmit the index of the selected candidate. From results obtained in previous sections, having two candidates is a good compromise. For further investigation, we retain two best configurations which correspond respectively for the use of one and two candidates for the SDec reference, and are designated by *SDec 1* and *SDec 2*.

We give the coding results of both *SDec 1* and *SDec 2* under all three configurations: AI, RA and LP, on the standard test set conforming to CTC then on an extended test set which includes additional sequences known for their challenging content. Different coding rates are also used to evaluate the performance of the proposed SDec scheme: low bit rate (LBR) and medium bit rate (MBR). This yields to two quality levels with average peak-signal-to-noise ratio for luminance of 33.5 dB and 36.4 dB respectively, corresponding to bitrates relevant for specific applications. Statistical analysis is eventually given. Our experiments are performed using HM12.

The test configuration can be summarized as follows:

Codec: HEVC, test model version 12.0 (HM12) Coding rates: LBR QP = {27, 32, 37, 42}, MBR QP = {22, 27, 32, 37} Test sequences: CTC test set for HEVC, with additional sequences of various resolutions Evaluation: Average B-D rate on all frames for configurations AI, RA and LP

3.3.1 Experimental results

The table 3.15 presents the experimental result of *SDec 1*.

Systematic gain is achieved for all tested sequences. On the HEVC test set, an average of -0.3%, -0.3% and -0.8% bit rate reduction for luminance are achieved respectively for AI, RA and LP configurations in MBR. Gain

Sequences class	AI		RA		LP	
	LBR	MBR	LBR	MBR	LBR	MBR
Class A	-0.2	-0.1	-0.3	-0.3	-1.8	-1.1
Class B	-0.3	-0.2	-0.3	-0.2	-1.3	-0.8
Class C	-0.4	-0.3	-0.4	-0.3	-1.0	-0.6
Class D	-0.3	-0.2	-0.3	-0.2	-0.9	-0.5
Class E	-0.7	-0.6	-0.5	-0.4	-1.3	-0.8
Class F	-0.4	-0.3	-0.5	-0.3	-1.4	-1.0
Average	**-0.4**	**-0.3**	**-0.4**	**-0.3**	**-1.3**	**-0.8**
Max gain	-1.1	-0.9	-1.0	-0.7	-3.2	-2.6
Book 2160p	-0.6	-0.4	-2.4	-1.9	-3.5	-2.5
QuadAndFly_part1 2160p	-0.8	-0.6	-1.6	-1.3	-2.1	-1.9
ParisFade 1088p	-0.2	-0.1	-14.9	-14.5	-16.9	-11.1
RollingTomatoes 1080p	-1.3	-0.7	-1.5	-1.5	-3.4	-3.5
DespicableMeMoon 1080p	-0.7	-0.6	-0.4	-0.5	-3.1	-2.1
RushHour 1080p	-0.3	-0.3	-0.7	-0.5	-2.9	-2.0
Average	**-0.7**	**-0.5**	**-3.6**	**-3.4**	**-5.3**	**-3.9**
Enc Time	292%		158%		158%	
Dec Time	379%		181%		216%	

Table 3.15 – *Bit rate savings (%) of SDec 1 (Ref: HM12)*

up to -2.6% is observed under LP configuration. We also observe that the proposed SDec scheme is more efficient in LBR.

On additional sequences, *SDec 1* performs very well, with an average of -0.5%, -3.4% and -3.9% of gain respectively in AI, RA and LP configurations in MBR, proving that the proposed SDec scheme suits particularly for sequences containing complex motion.

Encoding and decoding time are also increased quite significantly. This is mainly due to the competition of Intra directions on the SDec reference.

By adding a second candidate for the SDec reference, the result is given in table 3.16. Compared with *SDec 1*, a slightly higher average gain is obtained: -0.4%, -0.5%, -0.9% for the HEVC test set and -0.6%, -3.8%, -4.3% for the set of additional sequences respectively in AI, RA and LP configurations in MBR. This gain increase shows that a better prediction is achieved by using additional information provided by the second candidate. However, the runtime is further increased due to the SDec competition on the additional candidate block.

3.3.2 Statistical analysis

3.3.2.1 SDec selection rate

Let n_{SDec} be the number of PUs encoded in SDec mode and n_{tot} be the total number of encoded PUs. We define the SDec selection rate of as the percentage of PUs encoded in SDec mode:

$$\frac{n_{SDec}}{n_{tot}} \times 100\%$$

Table 3.17 shows the SDec selection rate of both versions *SDec 1* and *SDec 2* tested in MBR under configuration LP. Significant selection rates

Sequences class	*AI*		*RA*		*LP*	
	LBR	**MBR**	**LBR**	**MBR**	**LBR**	**MBR**
Class A	-0.2	-0.2	-0.5	-0.4	-2.1	-1.3
Class B	-0.5	-0.4	-0.6	-0.4	-1.5	-0.9
Class C	-0.5	-0.4	-0.7	-0.5	-1.3	-0.8
Class D	-0.4	-0.3	-0.5	-0.3	-1.0	-0.6
Class E	-0.8	-0.7	-0.6	-0.5	-1.3	-0.9
Class F	-0.8	-0.7	-0.8	-0.7	-1.5	-1.2
Average	**-0.5**	**-0.4**	**-0.6**	**-0.5**	**-1.5**	**-0.9**
Max gain	-1.2	-1.0	-1.5	-1.0	-3.5	-2.6
Book 2160p	-0.7	-0.5	-3.4	-2.6	-4.3	-3.2
QuadAndFly_part1 2160p	-1.0	-0.7	-2.7	-2.3	-3.1	-2.6
ParisFade 1088p	-0.3	-0.2	-15.6	-14.6	-17.0	-11.3
RollingTomatoes 1080p	-1.3	-0.9	-2.1	-1.9	-4.2	-4.2
DespicableMeMoon 1080p	-1.3	-1.0	-0.7	-0.7	-3.4	-2.3
RushHour 1080p	-0.3	-0.3	-0.7	-0.6	-3.1	-2.2
Average	**-0.8**	**-0.6**	**-4.2**	**-3.8**	**-5.9**	**-4.3**
Enc Time	468%		207%		208%	
Dec Time	944%		285%		324%	

Table 3.16 – *Bit rate savings (%) of SDec 2 (Ref: HM12)*

of 6.8% and 7.8% are obtained respectively, proving that proposed SDec mode can efficiently compete with existing coding modes.

Sequences class	*SDec 1*	*SDec 2*
Class A	11.4	12.8
Class B	6.7	7.8
Class C	6.6	8.0
Class D	4.6	5.4
Class E	2.5	2.7
Class F	8.9	9.9
Average	**6.8**	**7.8**

Table 3.17 – *Selection rate (%) of SDec mode for SDec 1 and SDec 2 in MBR under LP configuration.*

3.3.2.2 Coding modes replaced by SDec mode

Considering *SDec 1* tested in MBR under LP configuration, the graph in figure 3.12 gives, for each PU size, the percentage distribution of coding modes that are replaced by the newly introduced SDec mode. In other words, it answers the question: what is the next best coding mode in terms of R-D cost when SDec is the optimal? The result shows that for small blocks, the SDec mode is able to compete fairly well with all other coding modes. For example, for 21.3% of 8×8 blocks encoded in SDec mode, SDec performs even better than Skip which is a very efficient coding mode. For large 64×64 blocks, SDec mode can only compete with HEVC Intra since other coding modes exploiting motion correlation (Inter, Merge, Skip) are more efficient.

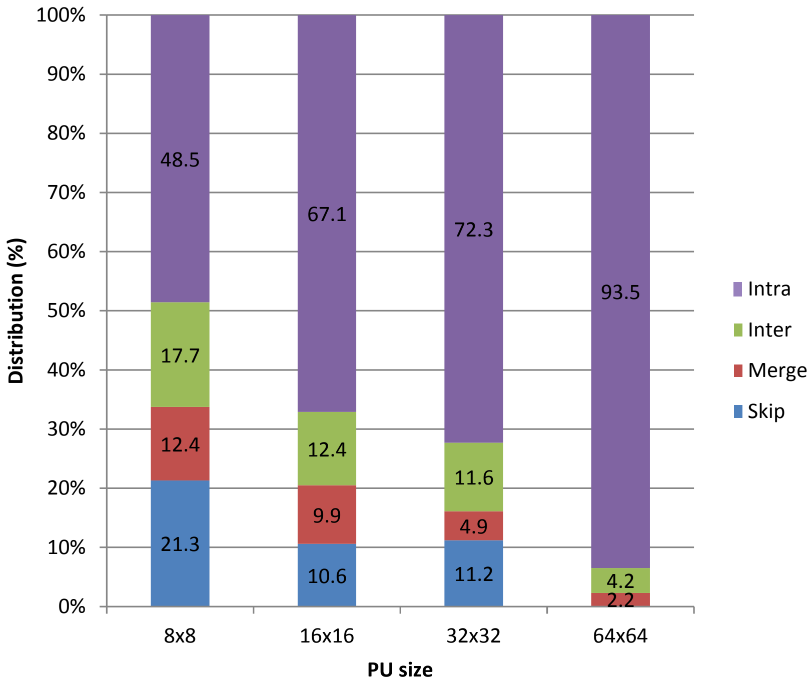


Figure 3.12 – *Distribution in percentage of classic coding modes replaced by SDec for SDec 1 in MBR under LP configuration.*

The observation made from the graph in figure 3.12 suggests a way to reduce the runtime by deactivating the classic Intra mode when SDec mode is selected on 64×64 blocks. The experiment provides a slight saving in runtime of 2% and a loss in average gain of 0.1%.

3.4 Perspectives

The practical scheme of SDec using only Intra as coding mode proposed in this chapter shows a promising result and confirms the performance of the whole SDec scheme. Several interesting perspectives can be considered to further improve the compression performance:

- Reduce the complexity for HEVC Intra coding mode, by removing the Intra direction already used for the SDec mode out of the list of tested directions. Indeed, signaling that Intra direction for the HEVC Intra mode requires more signaling overhead than when using SDec mode, which costs nothing as the Intra direction is inherited from the SDec reference.
- Additionally, it is proposed to improve the HEVC Intra MPM signaling mechanism by not including in the MPM list the Intra direction already used in the SDec mode. In this case, another direction can be inserted in the list, providing an improvement in gain if the newly inserted direction turns out to be the optimal Intra direction of the current block.

Both approaches require SDec mode to be tested before HEVC Intra mode during the competition of all coding modes on the current block because the result of the SDec mode is exploited to make improvements for the HEVC Intra mode.

Conclusion

Following the chapter presenting the general SDec scheme, a simplified practical application is proposed, using only Intra as coding mode for the SDec competition. Other parameters of the SDec scheme concerning the signaling of newly introduced syntax elements, the selection of candidates to be the SDec reference and the number of candidates to be used are also discussed.

Two most promising configurations using one and two candidates respectively are finally derived and taken for further analysis. It is demonstrated that an overall performance improvement is obtained despite different restrictions regarding the SDec general outline. On HEVC test set under CTC, the configuration using two candidates for the SDec reference yields an average bit rate savings of -0.4%, -0.5% and -0.9% respectively for AI, RA and LP configurations comparing to standard HEVC. Those obtained gains are still limited given the introduced complexity that is not negligible. Several perspectives are also given in order to reduce the complexity and to further improve the coding quality.

4 SDec based coding scheme using Intra 1D mode

In this chapter, we propose another practical application of the general SDec coding scheme proposed in the chapter 2, using Intra 1D as the coding mode for the SDec process. First, we detail our implementation of Intra 1D under HM12. Then, different intrinsic parameters of Intra 1D are tested to find the best configuration to be used in the SDec scheme. Finally, Intra 1D is used as a coding mode in the SDec scheme in order to reduce the signaling overhead. Experimental results under JCT-VC common test conditions are also given.

4.1 Implementation of Intra 1D

As already described in the chapter 1, Intra 1D is an Intra prediction mode with 1-dimensional partitioning. The main difference compared to the classic Intra mode can be summarized in the following characteristics that are exploited as intrinsic parameters of Intra 1D:

- the linear 1D-partition direction: Horizontal (HOR), Vertical (VER),
- the scan order of 1D-partitions: Raster, Bi-directional (Bi-Dir), Hierarchical (Hier),
- the Intra predictor for each linear 1D-partition.

In order to comply with the HEVC standard, we propose to use 35 Intra directions as predictors for each linear 1D-partition. Also, we keep using the conventional DCT transform instead of implementing the 1D DCT transform for the sake of simplification. Consequently, the residual content of a 1D-partition is arranged into square block so that HEVC DCT transform can be applied. Furthermore, Intra 1D will only be applied on luminance component.

The following syntax elements are introduced for signaling Intra 1D mode at the block level:

- $intra1D\_flag$: indicates if the current block is encoded with Intra 1D mode,
- $intra1D\_dir$: indicates the optimal 1D-partition direction (HOR or VER),
- $intra1D\_scan$: indicates the optimal scan order of 1D-partitions (Raster, Bi-Dir or Hier).

At 1D-partition level, the following additional syntax elements are transmitted:

- $intra1D\_dir\_luma$: indicates the optimal Intra direction for a 1D-partition,
- $intra1D\_cbf$: equals to 0 if all residual coefficients (after transformation and quantization) of a 1D-partition are zero, to 1 otherwise, in which case all the residual coefficients are transmitted in the bit stream.

It is important to note that, due to the re-arrangement process into square block for each 1D-partition, for a block of size 8×8 or 32×32, a 1D-partition, containing 8 or 32 pixels, is defined to contain two block lines, which yields in total 16 or 64 values. As such, we can still arrange the residual values of a 1D-partition into square block and apply existing HEVC DCT transform. The quantification process which follows the transformation is kept unmodified.

Figure 4.1 presents the signaling scheme of the Intra 1D mode with its syntax elements. The Intra 1D mode is introduced so that it competes with the classic Intra mode.

We remark that this signaling scheme of Intra 1D would not be efficient in case Intra 1D does not perform better than the classic Intra mode. The reason is that the signaling of the classic Intra mode is penalized by the introduced syntax element $intra1D\_flag$ as shown in the figure 4.1. How-

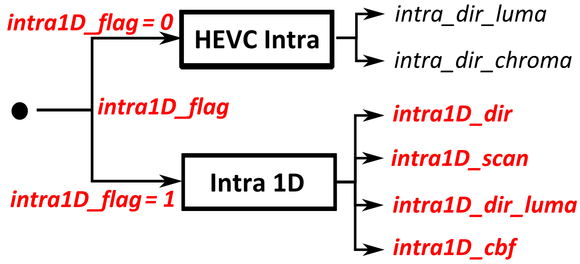


Figure 4.1 – *Signaling scheme for introduced Intra 1D syntax elements.*

ever, this signaling scheme will be removed later as Intra 1D is exploited as a coding mode in the SDec scheme.

4.2 Analysis on different parameters of Intra 1D

In this section, we present several experiments covering different intrinsic parameters of the Intra 1D coding mode. Experiments concerning the directions of 1D-partitions are first presented. Then, experiments to evaluate different scan orders are performed. Next, the number of Intra directions for each 1D-partition is investigated. Finally, statistical analysis is given to evaluate the impact of adding Intra 1D mode.

We conduct our experiments only in the AI configuration. The compression performance of all tests are evaluated using Bjøntegaard Delta (B-D) rate.

The following test configuration is used for all the experiments:

Codec: HEVC, test model version 12.0 (HM12) QPs range: Medium bit rate QP = {22, 27, 32, 37} Test sequences: CTC test set for HEVC, shortened to 5 frames Evaluation: Average B-D rate in AI configuration

4.2.1 Direction of 1D-partitions

We evaluate the contribution of each 1D-direction, respectively horizontal (HOR) and vertical (VER), to the overall coding performance of Intra 1D. For simplification purpose, we only use Raster scan for the processing order of 1D-partitions. All 35 available Intra directions are used as predictors for each 1D-partition. Table 4.1 evaluates the compression performance of implemented Intra 1D in different following cases:

– Only HOR direction is used. The syntax element *intra1D_dir* is then not signaled in the bit stream.
– Only VER direction is used. *intra1D_dir* is not signaled.
– Both HOR and VER directions are used in competition. *intra1D_dir* is signaled with one bit using CABAC with one context. The initial probability of the single CABAC model is set arbitrarily to 50% for simplification purposes since the probability of the model will converge during the encoding of a frame.

We observe that the use of both HOR and VER in competition yields the highest gain, proving that adding VER direction provides further coding improvement although it does not give any gain in average when be-

	HOR	VER	HOR & VER
Class A	0.0	0.0	0.0
Class B	0.0	0.0	0.0
Class C	-0.3	-0.1	-0.3
Class D	-0.1	0.0	-0.2
Class E	0.1	0.1	0.1
Class F	-0.9	-0.1	-1.1
Overall	**-0.2**	**0.0**	**-0.3**
Max	-2.6	-0.2	-2.7
Min	0.3	0.3	0.3
EncTime	568%	590%	1052%
DecTime	123%	126%	116%

Table 4.1 – *Coding performance of Intra 1D in function of the 1D-direction in used (Ref: HM12).*

ing used separately. The encoding time is doubled when using both the directions compared to the use of a single candidate. The decoding times in the three test cases are however nearly on the same level since only one direction is used by the decoder to decode a block encoded in Intra 1D mode. The gain in average is however small because Intra 1D mode is only efficient on some test sequence classes (C, D, F). A significant gain of -2.7% is observed on the sequence *SlideEditing*_1280×720.

By further analyzing the content of tested sequences, we observe that blocks encoded with Intra 1D horizontal or vertical often have texture in corresponding direction. Figure 4.2 shows the content of the sequences *PartyScene*_832×480 and *BQMall*_832×480 with blocks encoded in Intra 1D highlighted in yellow and red colors respectively for HOR and VER 1D-directions. It is also observed that Intra 1D mode is used to encode blocks with complex or non-linear texture that cannot be efficiently encoded using HEVC Intra mode. Figure 4.3 illustrates the content of the sequence *SlideEditing*_1280×720 which yields the highest gain of -2.7%. There are mostly small and complex typing letters that are very difficult to be precisely predicted by HEVC Intra mode, even by splitting into small blocks of size 8×8. On the contrary, Intra 1D performs efficiently on those particular content of sequence thanks to the finer block partitioning.

We also give the percentage of each 1D-direction when both HOR and VER 1D-directions are competing with each other in table 4.2.

QP	HOR	VER
22	70.2	29.8
27	68.9	31.1
32	65.2	34.8
37	64.7	35.3
Avg.	**67.8**	**32.2**

Table 4.2 – *Percentage (%) of each 1D-direction when both HOR and VER are used in competition with each other.*

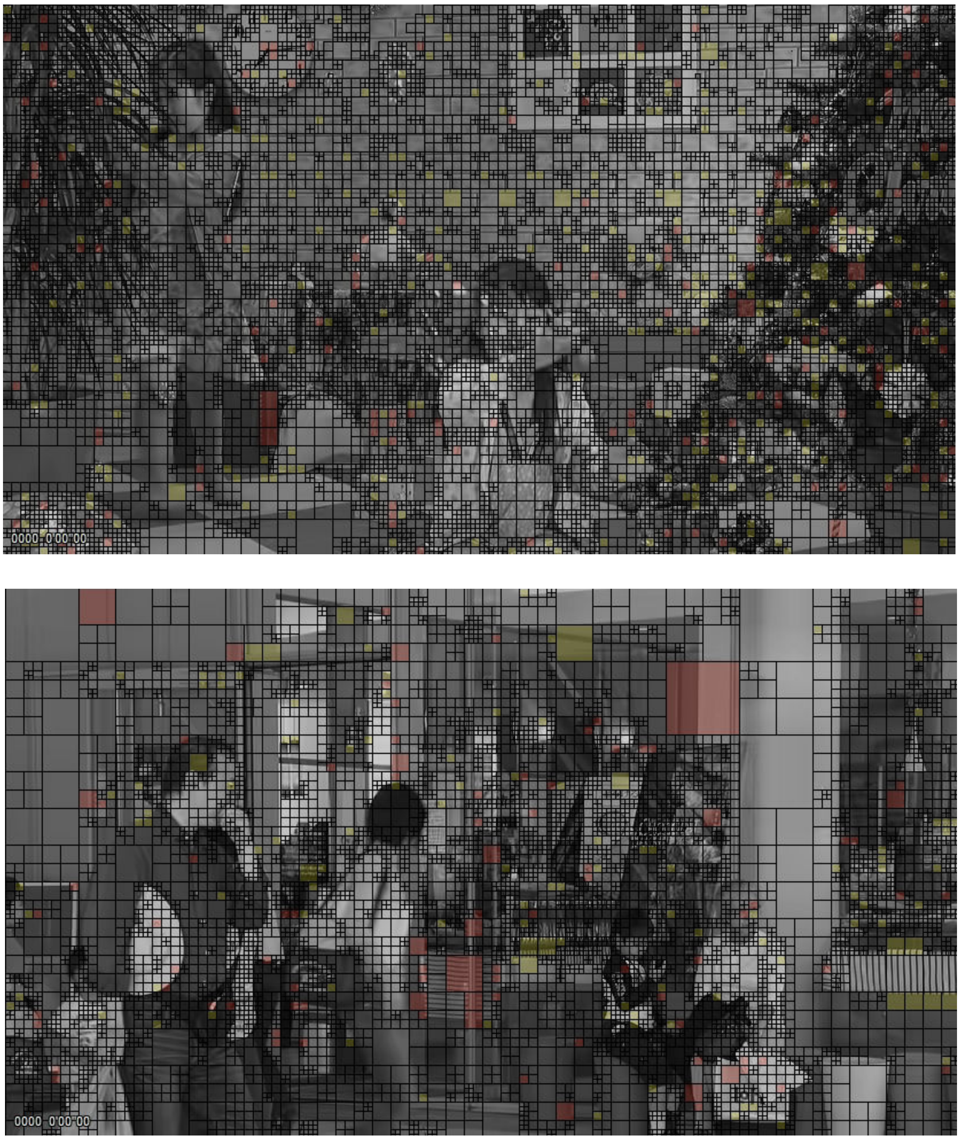


Figure 4.2 – *Blocks encoded with Intra 1D highlighted in yellow and red for HOR and VER directions in the sequences PartyScene*_832 × 480 *(above) and BQMall*_832 × 480 *(bottom).*

We observe that HOR direction contributes indeed much more than VER direction, with 67.8% of total blocks encoded in Intra 1D, resulting in a better coding gain if used separately.

4.2.2 Selection of scan order for 1D-partitions

The following experiment evaluates the contribution of different scan orders of 1D-partitions, respectively raster, bi-directional (Bi-Dir) and hierarchical (Hier) scans. We first conduct tests using each scan order separately. The syntax element *intra1D_scan* is not signaled in this case.

Experimental results are presented in table 4.3. We observe that only the raster scan gives coding gain in average. Bi-directional and hierarchical scans do not provide any gain in overall. This can be explained as follows: the Intra 1D mode is mostly selected on small blocks (8 × 8) according to the table 4.6 in section 4.2.4. Therefore, raster scan is often more efficient

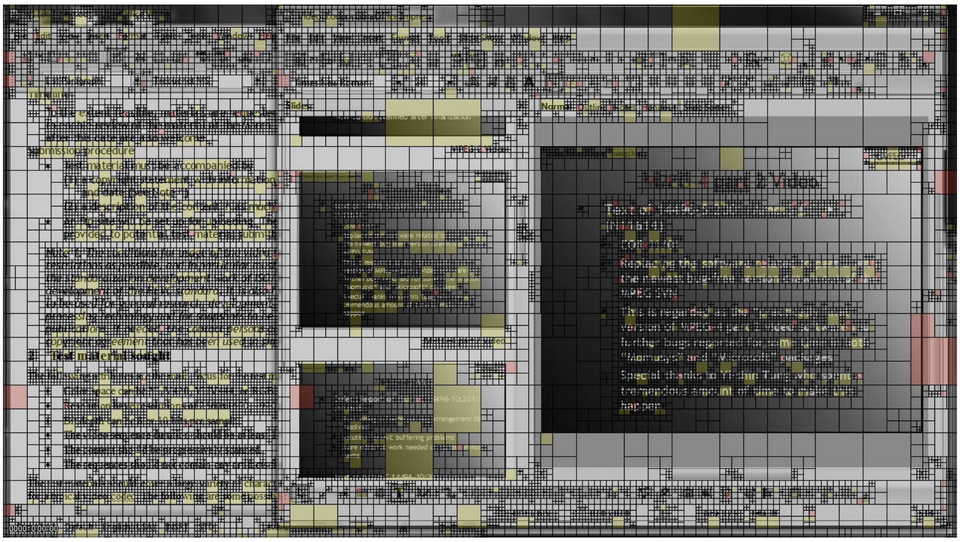


Figure 4.3 – *Blocks with complex non linear texture (typing letters) encoded with Intra 1D in the sequence SlideEditing_*1280×720.

since the texture variation in small blocks is more subtle than in large blocks.

Then, all three scan orders are put together in competition, which requires the signaling of *intra1D_scan* syntax element using up to two bits with one CABAC context for each bit as illustrated in figure 4.4. We remark that raster is privileged over two remaining scan orders given its higher performance observed during previous tests with separate scan order.

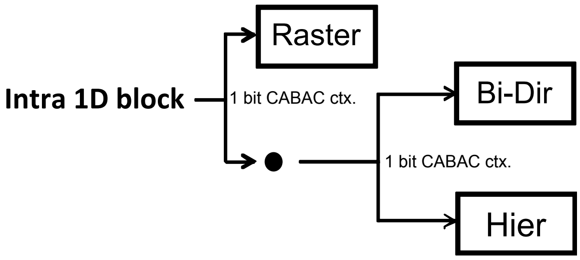


Figure 4.4 – *Signaling of intra1D_scan in case all 3 scan orders Raster, Bi-Dir and Hier are used.*

According to table 4.3, the combination of all three scan orders does not improve further the average gain of -0.3% obtained using uniquely the raster scan. Nevertheless, slight improvement in maximum gain is obtained.

Concerning the runtime, combining three scan orders increases indeed the encoding time by a factor of three. The decoding time is also slightly increased.

In case all three scan orders are used, we detail the percentage of each scan order in table 4.4 for all the blocks encoded in Intra 1D mode. According to table 4.4, raster scan order contributes the most in the performance of Intra 1D, with 89.3% of total blocks encoded in Intra 1D mode. This majority against bi-directional and hierarchical scans confirms again the

	Raster	Bi-Dir	Hier	All scans
Class A	0.0	0.0	0.0	0.0
Class B	0.0	0.1	0.0	0.0
Class C	-0.3	-0.1	-0.1	-0.3
Class D	-0.2	0.0	0.1	-0.3
Class E	0.1	0.2	0.1	0.0
Class F	-1.1	-0.3	0.0	-1.1
Overall	**-0.3**	**0.0**	**0.0**	**-0.3**
Max	-2.7	-0.9	-0.3	-2.8
Min	0.3	0.3	0.3	0.2
EncTime	1052 %	1290 %	1494 %	3513%
DecTime	116 %	124 %	121 %	123 %

Table 4.3 – *Compression performance of Intra 1D when different scan orders of 1D-partitions are used (Ref: HM12).*

efficiency of raster scan, which will be the unique scan order used in the next experiments.

QP	Raster	Bi-Dir	Hier
22	86.7	9.7	3.6
27	90.7	7.0	2.3
32	90.9	7.0	2.1
37	90.5	7.3	2.2
Avg.	**89.3**	**8.0**	**2.7**

Table 4.4 – *Percentage of each 1D-direction when both HOR and VER are used.*

4.2.3 Number of Intra directions for 1D-partitions

To reduce the complexity of Intra 1D mode which is very significant, we propose to limit the number of Intra directions as predictors for each 1D-partition. Instead of using all 35 available directions, we perform several tests where 19, 11 and 7 directions are used. In details, Planar and DC are retained. Among 33 angular Intra directions indexed from 2 to 35, depending on the limited number of directions, retained directions are equally distributed so that there is no bias in any direction. Figure 4.5 illustrates an example where 11 directions are exploited for Intra 1D.

Another method to reduce the number of Intra directions is to apply a preliminary test that computes the "pseudo" R-D cost of all 35 Intra directions by simply calculating the distortion between the original and the predicted 1D-partition, while the inverse transform and quantization processes are not conducted. Only the 8 best directions that minimize the "pseudo" R-D cost are selected for the real R-D competition on a 1D-partition.

All the experimental results are presented in the table 4.5. It is observed that using 11 Intra directions provides the best compromise between the compression ratio and the complexity, with an average gain of -0.2% and a maximum gain of -2.2%, while reducing more than half the complexity

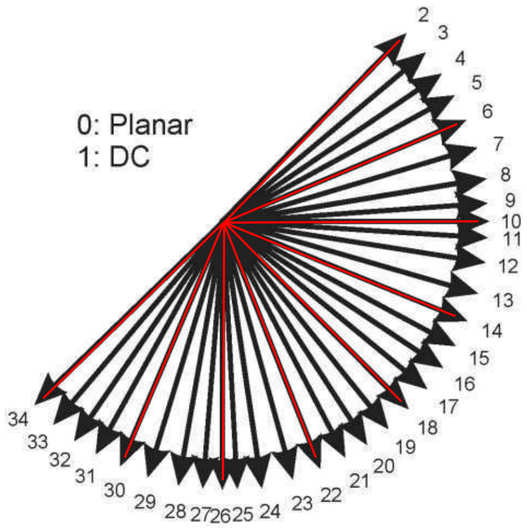


Figure 4.5 – *Reducing the number of Intra directions used as predictors for a 1D-partition from 35 to 11: only directions with indexes 0,1,2,6,10,14,18,22,26,30,34 are used.*

compared with the use of 35 directions. The test with 7 directions also yields -0.2% in average gain, but the maximum gain obtained in the sequence where Intra 1D performs most efficiently drops significantly to only -1.6%.

	35 dirs.	**19 dirs.**	**11 dirs.**	**7 dirs.**	**8 dirs. among 35**
Class A	0.0	0.0	0.0	0.0	0.0
Class B	0.0	0.0	0.0	0.0	0.0
Class C	-0.3	-0.3	-0.4	-0.3	-0.1
Class D	-0.2	-0.3	-0.2	-0.1	0.0
Class E	0.1	0.0	0.0	0.1	0.2
Class F	-1.1	-0.8	-0.8	-0.7	0.0
Overall	**-0.3**	**-0.2**	**-0.2**	**-0.2**	**0.0**
Max	-2.7	-2.3	-2.2	-1.6	-0.3
Min	0.3	0.1	0.1	0.1	0.4
EncTime	1052 %	626%	423%	314%	348%
DecTime	116 %	115 %	112 %	114 %	120%

Table 4.5 – *Intra 1D compression performance when using different number of Intra directions as predictors for each 1D-partition.*

The use of the preliminary test that selects 8 best Intra directions for the R-D competition does not yield any gain in average. This is probably due to the re-arrangement of a 1D-partition into a square block, perturbing therefore the evaluation of the "pseudo" R-D cost which uses a square-based Hadamard transform to evaluate the distortion.

4.2.4 Analysis of the impact of Intra 1D on HEVC Intra

We conduct additional experiments to provide statistical data related to the introduced Intra 1D mode. Comparison between the reference

HEVC and proposed method will be given in order to highlight the impact of adding this new mode.

For the reference HEVC, the distribution of Intra blocks relative to block sizes on all sequences in the HEVC test set (containing different resolutions varying from 416×240 to 2560×1600) is given in table 4.6.

QP	64×64	32×32	16×16	8×8
22	0.3	4.4	13.5	81.8
27	0.5	5.3	16.6	77.6
32	0.6	5.7	20.1	73.6
37	0.8	6.5	24.8	68.0
Avg.	**0.5**	**5.4**	**18.2**	**75.9**

Table 4.6 – *Distribution (%) of HEVC Intra mode in function of block sizes on the reference HM12.*

If Intra 1D is introduced, according to table 4.7, the average percentage of all the Intra 1D blocks is very small compared with HEVC Intra blocks, reaching only 2.73% of total blocks. This explains the low gain in average on the HEVC test set. It is also observed that most of Intra 1D blocks are of size 8×8.

QP	**HEVC Intra**				**Intra 1D**			
	64×64	32×32	16×16	8×8	64×64	32×32	16×16	8×8
22	0.34	4.39	13.38	78.60	0.01	0.04	0.22	3.02
27	0.45	5.28	16.51	74.94	0.02	0.04	0.25	2.49
32	0.61	5.66	19.97	71.27	0.03	0.04	0.23	2.19
37	0.81	6.53	24.53	66.05	0.03	0.04	0.22	1.79
Avg.	**0.53**	**5.37**	**18.07**	**73.30**	**0.02**	**0.04**	**0.23**	**2.43**
Sum	97.27				2.73			

Table 4.7 – *Distribution (%) of HEVC Intra and Intra 1D modes in function of block sizes for the proposed method.*

Adding Intra 1D coding mode in the competition with HEVC Intra requires more information to be transmitted from the encoder to the decoder. Table 4.8 gives the proportion of signaling cost required for modes and predictors of both HEVC Intra and Intra 1D in the reference HEVC and in our implementation when Intra 1D is enabled.

Compared with statistics obtained on the reference HEVC, we observe an overall increase of signaling overhead for the modes and the predictors, from 9.3% to 10.5% of the total bit stream. This is due to the signaling of several Intra 1D parameters on both PU and 1D-partition levels. Compared with HEVC Intra, given that only 2.73% of total blocks are encoded in Intra 1D, the proportion of signaling overhead dedicated for Intra 1D mode can be considered significant.

4.2.5 Best configuration

After conducting several experiments on different parameters of Intra 1D, we retain the best configuration for Intra 1D mode in terms of best

QP	Reference HEVC		Proposed method			
	Mode	Predictors	Mode		Predictors	
	HEVC Intra	HEVC Intra	HEVC Intra	Intra 1D	HEVC Intra	Intra 1D
22	0.8	6.2	0.8	0.5	5.9	0.7
27	1.2	8.3	1.1	0.7	7.6	1.0
32	1.7	10.5	1.7	0.9	9.7	1.4
37	2.5	13.6	2.3	1.2	12.5	1.8
Avg.	**1.2**	**8.1**	**1.2**	**0.7**	**7.6**	**1.0**
Sum	**9.3**		**10.5**			

Table 4.8 – *Percentage (%) of signaling cost in the total bit stream for modes and predictors regarding HEVC Intra and Intra 1D.*

compromise between the coding gain and the complexity, which includes following parameters:

- 1D-partition direction: both HOR and VER in competition,
- scan order for 1D-partitions: only Raster scan (*intra1D_scan* thus not signaled),
- predictors for each 1D-partition: 11 Intra directions.

The syntax element *intra1D_flag* is signaled using CABAC with three following contexts, with the initial probability set arbitrarily to 50%:

- None of Above and Left PU is encoded with Intra 1D mode,
- Above or Left PU is encoded with Intra 1D mode,
- Both Above and Left PU are encoded with Intra 1D mode.

We evaluate the coding performance under all three configurations: AI-Main, RA-Main and LD-P-Main, on the standard test set conforming to CTC then on an extended test set which includes additional sequences containing complex texture. Different coding rates (MBR and HBR) are also used.

The test configuration can be summarized as follows:

Codec: HEVC, test model version 12.0 (HM12) Coding rates: MBR QP = {22, 27, 32, 37}, HBR QP = {17, 22, 27, 32} Test sequences: CTC test set for HEVC, with additional sequences containing complex texture Evaluation: Average B-D rate on all frames in configurations AI, RA and LP

Experimental results are given in table 4.9. We observe that Intra 1D mode performs well only on sequences with complex texture in configuration AI. Furthermore, the amount of bit rate saving in HBR is higher than in MBR and in LBR because in high coding rate, textures are well preserved, allowing the proposed scheme to take benefit from the accurate prediction ability of Intra 1D. Figure 4.6 illustrates the content of some sequences with complex texture, where we can observe significant number of small details that cannot be predicted efficiently using HEVC Intra. Thanks to its finer block partitioning, Intra 1D is more suited to encode these sequences.

We also remark that, being an Intra mode, Intra 1D performs more efficiently on I-frames than on P- or B-frames. Indeed, it is observed on HEVC

Sequences class	AI		RA		LP	
	MBR	HBR	MBR	HBR	MBR	HBR
Class A	0.0	0.0	0.0	0.0	0.0	0.0
Class B	0.0	0.0	0.0	0.0	0.0	0.0
Class C	-0.2	-0.2	-0.1	-0.1	0.0	0.0
Class D	-0.2	-0.2	-0.1	0.0	0.1	0.0
Class E	0.0	0.0	-0.1	0.0	-0.1	0.0
Class F	-1.2	-1.2	-1.0	-0.9	-0.7	-0.6
Average	**-0.3**	**-0.3**	**-0.2**	**-0.2**	**-0.1**	**-0.1**
Max gain	-3.0	-3.0	-2.6	-2.5	-1.7	-1.7
BasketBall 1080p	-3.2	-6.4	-0.6	-1.0	-0.2	-0.7
S8FootFranceRou 1080p	-4.3	-7.1	-0.8	-1.0	-0.1	-0.3
S11FootLyonPSG 1080p	-2.4	-4.3	-0.9	-0.9	-0.1	-0.2
S15Rugby 1080p	-4.8	-10.3	-0.7	-1.3	-0.2	-1.0
S16DavisCup 1080p	-2.6	-5.0	-0.9	-1.0	0.0	-0.1
S18DavisCup 1080p	-2.1	-6.5	-0.7	-1.1	0.0	-0.2
Average	**-3.2**	**-6.6**	**-0.8**	**-1.0**	**-0.1**	**-0.4**
Enc Time	292%		158%		158%	
Dec Time	379%		181%		216%	

Table 4.9 – *Bit rate savings (%) of Intra 1D (Ref: HM12)*

sequences test set and additional sequences that the gain in configuration AI is much more significant than in RA, while nearly no gain is obtained in LP. It can be explained by the proportion of I-frames compared to P- and B-frames in each configuration. Furthermore, as the signaling overhead of the Intra 1D mode is too costly for configuration LP where a very light bit stream is maintained by exploiting the temporal redundancy between P-frames and by using efficient temporal coding modes such as Skip or Merge. On the contrary, for configurations AI and RA where several I-frames are used, Intra 1D mode provides an efficient spatial prediction capacity which can compensate for its significant signaling overhead, resulting eventually in an overall coding improvement.

4.3 Intra 1D used as a coding mode in the SDec scheme

4.3.1 Description

By definition, the Intra 1D mode requires a significant signaling overhead due to its large number of intrinsic parameters. In order to reduce this signaling overhead and to attempt to improve the coding gain, we apply the SDec scheme that exploits Intra 1D in the SDec competition. Figure 4.7 illustrates in details the proposed specific SDec scheme using Intra 1D coding mode. All intrinsic parameters of Intra 1D (partitioning direction, 1D-partitions scan order, predictors for each 1D-partition) are tested on the SDec reference by competition to find the Intra 1D configuration that minimizes the R-D cost. This optimal Intra 1D configuration, computed on the SDec reference, is inherited to encode the current block.

We propose that the syntax element $sdec\_flag$ signaling SDec mode is inserted in the signaling scheme of HEVC coding modes as represented in

Figure 4.6 – *Sequences containing complex texture suited for Intra 1D mode: from the top, BasketBall_1080p, s8FootFranceRou_1080p and s18DavisCup_1080p.*

figure 4.8. Compared with figure 4.1 representing the signaling scheme of Intra 1D, the use of the SDec scheme allows to skip the transmission of all the syntax elements related to Intra 1D competing parameters (*intra*1D_*dir*, *intra*1D_*scan* and *intra*1D_*dir*$_{l}$*uma*). The amount of bits dedicated for signaling the mode is therefore reduced.

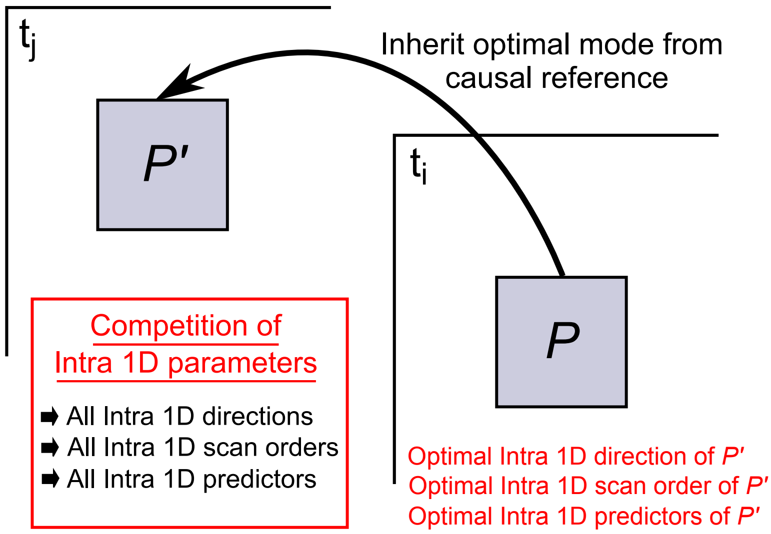


Figure 4.7 – *SDec coding scheme using Intra 1D mode.*

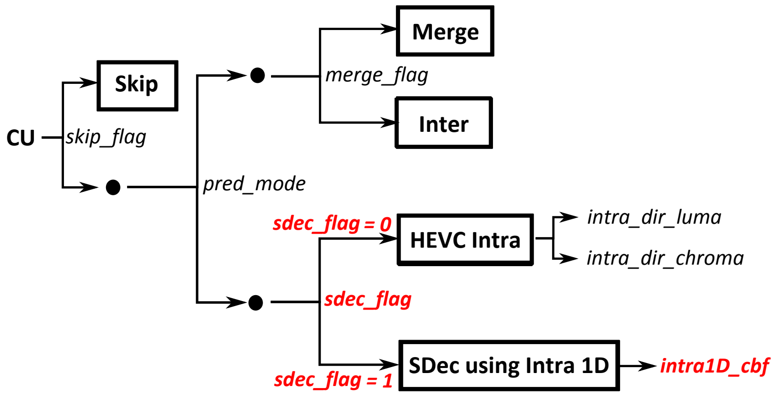


Figure 4.8 – *Signaling scheme for the introduced SDec syntax elements in the SDec scheme using Intra 1D mode.*

4.3.2 Parameters related to the SDec reference

We propose to evaluate the optimal number of candidates for the SDec reference that provides the best compromise between maximizing the coding gain while keeping the complexity within an acceptable limit. We conduct tests in configurations LP and AI, with different number of candidates as follows:

- **1 candidate**:
  - I-frames: Above block
  - P-frames: Colocated block
- **2 candidates**:
  - I-frames: Above and Left blocks
  - P-frames: Colocated and Above blocks
- **3 candidates**:
  - I-frames: Above, Left and AboveLeft blocks
  - P-frames: Col, Above and AboveRight blocks
- **4 candidates**:
  - I-frames: Above, Left, AboveLeft and AboveRight blocks
  - P-frames: Col, Above, AboveRight and AboveLeft blocks

All proposed block candidates are pre-identified blocks for simplification purposes. For each of the four cases corresponding to the number of candidates being exploited, the signaling method of the selected candidate is the same as in section 3.2.4.

Table 4.10 presents the experimental results in LP configuration:

	1 cand.	**2 cands.**	**3 cands.**	**4 cands.**
Class A	-0.1	-0.1	-0.1	-0.2
Class B	-0.1	-0.2	-0.2	-0.2
Class C	0.0	0.0	-0.1	-0.1
Class D	0.1	0.1	0.0	0.0
Class E	0.0	-0.2	-0.1	0.0
Class F	-1.4	-0.6	-1.5	-1.0
Average	**-0.3**	**-0.2**	**-0.3**	**-0.2**
Max gain	-3.0	-2.5	-2.9	-2.2
EncTime	202 %	292 %	385%	529 %
DecTime	198 %	256 %	322 %	415 %

Table 4.10 – *Compression performance of the SDec scheme using Intra 1D, with different number of candidates to be the SDec reference, LP configuration (Ref: HM12).*

Experimental results in AI configuration is provided in table 4.11, with additional sequences containing complex texture:

	1 cand.	**2 cands.**	**3 cands.**	**4 cands.**
Class A	0.0	0.0	0.0	0.0
Class B	0.0	-0.1	-0.1	-0.1
Class C	0.0	-0.1	-0.1	-0.1
Class D	0.0	0.0	0.0	-0.1
Class E	-0.1	-0.2	-0.2	-0.2
Class F	-0.3	-0.8	-0.8	-0.8
Average	**-0.1**	**-0.2**	**-0.2**	**-0.2**
Max gain	-0.7	-2.0	-2.0	-2.1
BasketBall 1080p	-0.6	-1.2	-1.4	-1.5
S8Francerou 1080p	-0.6	-1.5	-1.5	-1.6
S11FootLyonPSG 1080p	-0.3	-1.0	-1.0	-1.1
S15Rugby 1080p	-1.0	-2.0	-2.1	-2.4
S16DavisCup 1080p	-0.4	-1.0	-1.2	-1.3
S18DavisCup 1080p	-0.6	-0.9	-1.0	-1.1
Average	**-0.5**	**-1.3**	**-1.4**	**-1.5**
EncTime	392 %	698 %	981%	1181 %
DecTime	244 %	465 %	650 %	803 %

Table 4.11 – *Compression performance of the SDec scheme using Intra 1D, with different number of candidates to be the SDec reference, AI configuration (Ref: HM12).*

According to both tables 4.10 and 4.11, we observe that the runtime is increased proportionally to the number of candidates for the SDec reference. By considering a good compromise between coding gain and complexity, on HEVC test set and additional sequences, there is no interest in considering a number of candidates higher than two.

For the choice of the SDec reference block, as mentioned in previous chapter on the general outline of the SDec scheme (cf. 2.4), different coding modes require different criteria when evaluating blocks to be the SDec reference. Given that Intra 1D provides a better prediction accuracy than HEVC Intra, the SDec reference is better to be as similar to the current block as possible. Therefore, template matching technique is proposed to be used to identify the SDec reference instead of simply using a pre-identified block such as the colocated block.

Another test is thus conducted in configuration LP to evaluate the efficiency of the SDec reference block found by using the template matching method (cf. 3.2.2). The template thickness e and the search radius r are both set to 8 pixels since this configuration yields the best compromise between coding gain and complexity according to our test on different sets of (e, r). Both the results, respectively when using template matching method and when using the colocated block as the single SDec reference, are presented in the table 4.12 for comparison purpose. Tested sequences are encoded in 5 frames and the gain is calculated on P-frames only.

	Template matching	**Colocated block**
BasketballDrillText_832 $\times$ 480	-0.3	-0.7
ChinaSpeed_1024 $\times$ 768	0.6	0.0
SlideEditing_1280 $\times$ 720	-5.4	-1.1
SlideShow_1280 $\times$ 720	-0.4	0.5
Overall	**-1.4**	**-0.3**
EncTime	1270 %	240%

Table 4.12 – *Intra 1D compression performance when using the template matching technique to find the SDec reference in comparison with using the colocated block, configuration LP (Ref: HM12).*

We observe that for the sequence "SlideEditing_1280 $\times$ 720", the use of the template matching technique to find the SDec reference provides a significant improvement in coding performance over the use of the pre-identified colocated block. Indeed, since this particular sequence contains a texture which is complex and is in a translational motion (PC screen that displays typing letters and is scrolling up and down), the template matching process allows to efficiently find a SDec reference block that is identical to the current block. The Intra 1D parameters computed on the SDec reference is thus accurate to be inherited to encode the current block. On other tested sequences that do not contain complex texture in translational motion, the advantage of using template matching is diminished. We also observe a significant increase in runtime when exploiting template matching technique (1270% compared to 240% when using the pre-identified colocated block). This experiment proves again the importance of the selection of the SDec reference block that could make significant impact on the performance of the SDec scheme. In the next section, we will continue to use the colocated block as the SDec reference for further tests under LP configuration since it remains a versatile block candidate which can perform well on most sequences of the standard HEVC test set. A comparison between the use of the colocated block and the template matching technique applied on a large set of tested sequences will be also given.

4.3.3 Selected SDec configurations

To further investigate the performance of the SDec scheme using Intra 1D mode, we consider two following configurations, having respectively one and two candidates for the SDec reference, for additional experiments:

<u>SDec 1</u>: only one candidate for the SDec reference (*sdec_ref* not signaled)
– For I-frames: Above block
– For P- and B-frames: Colocated block

<u>SDec 2</u>: two pre-identified candidates for the SDec reference (*sdec_ref* signaled as index of the selected candidate)
– For I-frames: Above and Left blocks
– For P- and B-frames: Colocated and Above blocks

The syntax element *sdec_flag* is signaled using CABAC with three following contexts, with the initial probability set arbitrarily to 50%:
– None of Above and Left PU is encoded with SDec mode,
– Above or Left PU is encoded with SDec mode,
– Both Above and Left PU are encoded with SDec mode.

We first give the coding result of *SDec 1*, then of *SDec 2*. Experiments are conducted conforming to CTC, under all three configurations: LP, RA and AI. The standard test set conforming to CTC is used. Additional sequences containing complex texture are also tested. Furthermore, different coding rates are considered to evaluate the performance of proposed SDec scheme.

The test configuration can be summarized as follows:

Codec: HEVC, test model version 12.0 (HM12) QPs range: LBR QP = {27, 32, 37, 42}, MBR QP = {22, 27, 32, 37}, HBR QP = {17, 22, 27, 32} Test sequences: CTC test set for HEVC, with additional sequences containing complex texture Evaluation: Average B-D rate on all frames in configurations AI, RA and LP

Table 4.13 presents the coding performance of *SDec 1*. According to the table, systematic gain is observed. On HEVC test set in MBR, the average gains obtained in configurations AI, RA and LP are respectively -0.1%, -0.1% and -0.2%. On extended test set, average gains of -0.5%, -0.2% and -0.1% are observed under those three configurations. There are less gains in average in HBR than in MBR or LBR, except for AI configuration with additional test sequences.

If the SDec reference is found by exploiting the template matching technique, better compression performance is indeed obtained for some particular sequences that contain complex texture in a translational move-

Sequences class	AI			RA			LP		
	LBR	MBR	HBR	LBR	MBR	HBR	LBR	MBR	HBR
Class A	0.0	0.0	0.0	0.0	0.0	0.0	-0.4	-0.2	-0.1
Class B	-0.1	0.0	0.0	-0.1	0.0	0.0	-0.3	-0.1	0.0
Class C	-0.1	0.0	0.0	0.0	0.0	0.0	-0.1	0.0	0.0
Class D	0.0	0.0	0.0	0.0	0.0	0.0	-0.1	0.0	0.0
Class E	-0.2	-0.1	0.0	-0.2	-0.1	0.0	-0.3	-0.1	-0.1
Class F	-0.4	-0.3	-0.2	-0.5	-0.4	-0.3	-1.3	-1.0	-0.7
Average	**-0.1**	**-0.1**	**0.0**	**-0.1**	**-0.1**	**-0.1**	**-0.4**	**-0.2**	**-0.1**
Max gain	-0.7	-0.7	-0.7	-0.7	-0.6	-0.5	-3.0	-2.4	-1.9
BasketBall 1080p	-0.3	-0.6	-1.9	-0.5	-0.2	-0.2	-0.6	-0.3	-0.1
S8FootFranceRou 1080p	-0.2	-0.6	-1.8	-0.1	-0.1	-0.1	-0.1	0.0	0.0
S11FootLyonPSG 1080p	-0.1	-0.3	-1.9	-1.1	-0.6	-0.2	-0.6	-0.2	0.0
S15Rugby 1080p	-0.2	-1.0	-3.6	-0.9	-0.2	-0.2	-0.4	-0.1	0.0
S16DavisCup 1080p	-0.2	-0.4	-1.3	-0.1	-0.1	-0.1	-0.3	0.0	0.0
S18DavisCup 1080p	-0.1	-0.3	-1.9	-0.1	-0.1	-0.2	0.0	0.0	0.0
Average	**-0.2**	**-0.5**	**-1.9**	**-0.5**	**-0.2**	**-0.2**	**-0.3**	**-0.1**	**0.0**
Enc Time	392%			190%			189%		
Dec Time	244%			176%			202%		

Table 4.13 – *Bit rate savings (%) of SDec 1 (Ref: HM12)*

ment, as shown in table 4.14 which provides the gain compared to the use of the colocated block as the SDec reference.

Sequences class	LP		
	LBR	MBR	HBR
Class A	0.3	0.1	0.0
Class B	0.1	0.0	-0.1
Class C	-0.5	0.0	0.1
Class D	0.2	0.2	0.0
Class E	-0.3	-0.2	0.2
Class F	-0.6	-1.0	-1.2
Average	**-0.1**	**-0.1**	**-0.2**
Max gain	-7.7	-4.3	-4.0
BasketBall 1080p	0.2	0.2	-0.1
S8FootFranceRou 1080p	-0.6	-0.3	-0.3
S11FootLyonPSG 1080p	-0.2	0.0	0.0
S15Rugby 1080p	1.2	0.3	-0.2
S16DavisCup 1080p	0.2	0.2	0.0
S18DavisCup 1080p	-1.1	-0.3	-0.3
Average	**-0.1**	**0.0**	**-0.1**
Enc Time	514%		

Table 4.14 – *Bit rate savings (%) when exploiting the template matching technique to find the SDec reference compared to the use of the colocated block (Ref: SDec 1)*

We observe in table 4.14 that for sequences such as $''SlideEditing\_1280 \times 780''$ in the class F of the HEVC test set or $''S8FootFranceRou\_1080p''$ and $''S18DavisCup\_1080p''$ of the additional test set, significant improvement is achieved. In other sequences which do not contain this particular characteristic, no systematic gain is provided with the use of the template matching technique. We also remark that the runtime is increased roughly by five times compared to the use of the colocated block as the SDec reference.

Experimental results of *SDec 2* is given in table 4.15. Compared with *SDec 1*, exploiting a second candidate for the SDec reference indeed improves the coding performance. A systematic increase in gain is observed. On HEVC test set in MBR, the average gains obtained in configurations AI, RA and LP are respectively -0.2%, -0.1% and -0.3%. Average gains of -1.3%, -0.3% and -0.1% are observed on extended test set under those same configurations.

Sequences class	*AI*			*RA*			*LP*		
	LBR	**MBR**	**HBR**	**LBR**	**MBR**	**HBR**	**LBR**	**MBR**	**HBR**
Class A	-0.1	0.0	0.0	-0.1	0.0	-0.1	-0.5	-0.2	-0.1
Class B	-0.2	-0.1	0.0	-0.1	0.0	0.0	-0.3	-0.1	-0.1
Class C	-0.1	-0.1	-0.1	-0.1	-0.1	0.0	-0.1	0.0	0.0
Class D	-0.1	0.0	0.0	0.0	0.0	0.0	-0.1	0.0	0.0
Class E	-0.3	-0.2	-0.1	-0.2	-0.1	-0.1	-0.4	-0.2	-0.1
Class F	-1.1	-0.8	-0.7	-1.0	-0.7	-0.5	-1.4	-1.0	-0.7
Average	**-0.3**	**-0.2**	**-0.1**	**-0.2**	**-0.1**	**-0.1**	**-0.5**	**-0.3**	**-0.2**
Max gain	-2.4	-2.0	-1.9	-2.1	-1.7	-1.6	-3.5	-2.8	-2.1
BasketBall 1080p	-0.7	-1.2	-2.1	-0.6	-0.3	-0.3	-0.7	-0.3	-0.2
S8FootFranceRou 1080p	-0.4	-1.5	-3.5	-0.1	-0.1	-0.2	-0.1	-0.1	0.0
S11FootLyonPSG 1080p	-0.7	-1.0	-1.9	-1.3	-0.7	-0.4	-0.7	-0.2	-0.1
S15Rugby 1080p	-0.5	-2.0	-6.0	-0.9	-0.2	-0.3	-0.5	-0.1	0.0
S16DavisCup 1080p	-0.9	-1.0	-2.6	-0.4	-0.2	-0.3	-0.2	0.0	0.0
S18DavisCup 1080p	-0.4	-0.9	-3.6	-0.2	-0.2	-0.4	0.0	-0.1	-0.1
Average	**-0.6**	**-1.3**	**-3.5**	**-0.6**	**-0.3**	**-0.3**	**-0.4**	**-0.1**	**-0.1**
Enc Time	698%			274%			272%		
Dec Time	465%			226%			261%		

Table 4.15 – *Bit rate savings (%) of SDec 2 (Ref: HM12)*

If we compare tables 4.13 and 4.15 related to the SDec scheme using Intra 1D and table 4.9 related to the use of Intra 1D without the SDec scheme, it is observed that using the SDec scheme brings more gains uniquely under LP configuration. More specifically, it is only on P-frames that the SDec scheme using Intra 1D can further improve the gain compared with the Intra 1D mode. The reason can be explained by the use of a more suitable candidate to be the SDec reference on P-frames, allowing to exploit more efficiently the temporal redundancy than the spatial redundancy. Further details are given in the next section which provides different statistical analysis.

4.3.4 Statistical analysis

We conduct additional tests to provide some statistics useful for our analysis. Table 4.16 shows the SDec selection rate of *SDec 2* tested in MBR on the HEVC test set under AI configuration in function of block sizes. Among all blocks, 3.3% are encoded with the SDec mode. We observe that most blocks encoded in SDec using Intra 1D are of size 8×8 (77.5%), followed by size 16×16 (15.6%), 32×32 (5.3%) and 64×64 (1.7%). This observation provides a possible way to reduce the complexity of the proposed SDec scheme, by deactivating for example the SDec mode on blocks with size of 32×32 or 64×64.

QP	*%SDec*	8×8	16×16	32×32	64×64
22	2.2	82.1	12.4	4.7	0.8
27	3.2	81.2	12.7	4.7	1.4
32	3.8	76.3	16.5	5.3	1.9
37	4.7	71.6	19.7	6.4	2.3
Avg.	**3.3**	**77.5**	**15.6**	**5.3**	**1.7**

Table 4.16 – *Selection rate (%) of SDec using Intra 1D mode and distribution (%) of blocks encoded in SDec mode in function of block sizes.*

Considering *SDec 1* tested in MBR with configuration LP, table 4.17 gives the distribution of HEVC coding modes that are replaced by the newly introduced SDec mode using Intra 1D. In other terms, it indicates the second best coding mode in terms of R-D cost when SDec is the optimal mode.

QP	Skip	Merge	Inter	Intra
22	16.0	26.3	30.0	27.7
27	15.2	27.4	30.9	26.5
32	20.1	25.9	35.4	18.6
37	26.5	23.0	31.4	19.1
Avg.	20.1	25.5	32.4	22.0

Table 4.17 – *Distribution (%) of HEVC coding modes replaced by the SDec mode for SDec 1 in MBR under LP configuration.*

According to the table 4.17, when the SDec mode using Intra 1D is selected, it replaces the Skip and Merge modes respectively on 20.1% and 25.5% of blocks encoded in SDec mode. This observation proves that in LP configuration, the SDec scheme using Intra 1D can efficiently exploit temporal redundancy to provide accurate prediction that can compete with modes such as Skip and Merge, which are considered as very efficient coding modes that do not require significant amount of signaling overhead. In consequence, under LP configuration and for P-frames, the proposed SDec scheme using Intra 1D can improve further the coding gain compared with the use of only Intra 1D. On the contrary, for I-frames, the spatial redundancy exploited from the SDec reference is probably not correlated enough to provide efficient coding performance, resulting therefore in less coding gain under AI and RA configurations for the SDec scheme using Intra 1D compared with the use of only Intra 1D.

It is also important to note that for the SDec scheme using Intra 1D, a slight distortion between the SDec reference and the current block can have a significant negative impact on the coding performance due to the fine precision of Intra 1D in inheriting characteristics of the SDec reference.

Therefore, it is better to exploit temporal redundancy which generally provides better correlation than spatial redundancy.

4.4 Perspectives

There are some interesting perspectives which can be considered to further improve the compression performance:

- For I-frames, exploit the template matching technique to find a block to be the SDec reference in the causal region of the current frame. This could improve the performance of the proposed method in AI and RA configurations, especially for sequences containing complex and repetitive pattern.
- Given that Intra 1D works often on small blocks (8×8), it is possible to reduce the complexity by deactivating the SDec scheme using Intra 1D mode on blocks with size of 32×32 or 64×64 for example.

Conclusion

In this chapter, an implementation of the Intra 1D coding mode in HEVC is proposed. Different intrinsic parameters of Intra 1D are tested to find the optimal configuration that provides the best compromise between coding gain and complexity. Experimental shows that Intra 1D is efficient uniquely for coding sequences containing complex texture that cannot be easily predicted using HEVC Intra, with average gain of -3.2%, -0.8% and -0.1% respectively in AI, RA and LP configurations on a selected test set. On the HEVC test set, average gains of -0.3%, -0.2% and -0.1% are obtained respectively in those three configurations.

Next, Intra 1D is used as a coding mode in the SDec scheme, resulting in another practical application of the SDec general outline already presented in the chapter 2. Preliminary tests related to the block candidates for the SDec reference are given, allowing to eventually consider two configurations which exploit respective one and two candidates. Experimental results prove that, compared with Intra 1D, the SDec scheme using Intra 1D can improve further the gain but uniquely in configuration LP where the temporal redundancy is exploited more efficiently. In AI and RA configurations where the contribution of I-frames is much more significant, the spatial redundancy exploited from the SDec reference is not relevant enough to encode the current block, resulting in a reduced performance compared with the case of using only Intra 1D. Experimental results show that on the sequences containing complex texture, the configuration using two candidates for the SDec reference yields an average bit rate savings of -1.3%, -0.3% and -0.1% respectively for AI, RA and LP comparing to standard HEVC. Average gains of -0.2%, -0.1% and -0.3% are observed on the HEVC test set respectively in those three configurations.

Furthermore, it is proved that using more a sophisticated method to select the SDec reference, such as the template matching technique applied in previously decoded frames, provides gain improvement on particular sequences that contain complex texture in translational movement. In order to further improve the performance, several perspectives can be considered, such as exploiting the template matching technique on causal region of the current frame to find the SDec reference, or reducing the complexity by deactivating the proposed scheme on blocks with specific sizes.

5 SDec based coding scheme inheriting re-estimated motion parameters

In this chapter, a new practical application of the SDec scheme is proposed. The aim is to exploit the motion parameters that are re-estimated on the causal SDec reference and inherit them to encode the current block. First, the general description of the proposed scheme is given. Its advantages and drawbacks are then discussed. Next, a first practical application which uses motion re-estimation based on block matching technique is presented, including the detailed description, the experimental results and the statistical analysis. After that, a second version which re-estimates motion parameter by exploiting Optical Flow (OF) technique is given. This second version is applied in both 2D and 3D coding. For each type of coding, the description and the experimental results are detailed.

5.1 General description

According to the general description of the SDec scheme, any coding mode can be tested on the SDec reference so that the optimal mode can be inherited later to encode the current block. We thus propose the temporal coding mode to be used for the SDec process. Following the SDec principle, the idea consists to re-estimate the motion parameter of the SDec reference block by re-encoding the block during the SDec process and inherits the computed motion parameter to encode the current block.

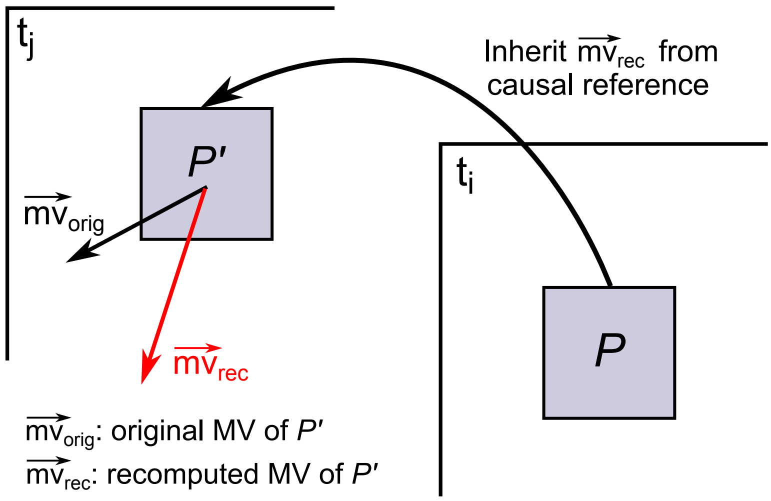


Figure 5.1 – *SDec scheme inheriting motion parameters re-estimated on SDec reference.*

As illustrated in figure 5.1, let us suppose that we are encoding the current block P in the frame at instant t_i. A reconstructed region P' in an already reconstructed frame at instant t_j is considered as the SDec reference. A motion estimation is conducted on the SDec reference P' to compute its optimal motion parameter. Different techniques can be used to find the motion parameter pointing to a region that best describes the reconstructed P'. We can cite for example the classic block matching based motion estimation or the complex technique of warped optical flow (WOF) which provides a dense motion vectors field (DMVF) instead of a motion vector (MV).

If P' is already encoded with a temporal coding mode, it has a MV denoted by $\overrightarrow{mv}_{orig}$. Its new motion parameter, recomputed by the SDec process, is denoted by $\overrightarrow{mv}_{rec}$ and can be similar to $\overrightarrow{mv}_{orig}$. $\overrightarrow{mv}_{rec}$ is eventually inherited to encode the current block P without being transmitted in the bit stream, saving thus the signaling overhead.

Is it important to note that the original MV was computed based on data of the original block, while the re-estimated MV is obtained based uniquely on data of the reconstructed block.

5.2 Advantages and drawbacks

The main advantage of the method is the possibility to inherit motion parameters from the SDec reference without the need to signal them in the bit stream, allowing in consequence a reduction in signaling overhead. In this way, it is similar to the Merge coding mode since both methods inherit motion parameters from a causal block for the current block. However, the major difference remains in the process with which MVs are computed. Merge method inherits the original MV which is computed using the R-D criterion since the signaling of the MV must be taken into account (transmitting the difference in X and Y components or simply using the Merge index), while the proposed scheme re-estimates the new motion parameter using uniquely the distortion criterion since no signaling cost for the MV is needed given that the entire re-estimation process exploits only the reconstructed data.

The drawback is that the re-estimation of motion parameter is based on the reconstructed data of the causal SDec reference block, inducing the unavoidable error on the re-estimated MV unlike the original MV which was computed based on the original data of the block. Another drawback is the complexity required for re-estimating the best motion parameter on the SDec reference.

5.3 Implementation with motion parameters re-estimated using block matching technique

5.3.1 Description

In this section, we present a practical application of the SDec scheme inheriting the motion parameters. It is proposed that new motion parameters of the SDec reference are re-evaluated using the HEVC motion estimation process which is based on the block matching technique. More precisely, the entire motion estimation conventionally used to find the motion parameters of the current block in Inter mode is exploited to find the motion parameters of the SDec reference, with the exception that only block distortion is used as evaluating criterion.

We consider the block candidates used for the construction of the Merge list as the candidates for the SDec reference. More precisely, each Merge MV candidate inserted in the Merge list belongs to a causal block that covers a specific position as illustrated in figure 5.2. All five candidates of the Merge list are extracted among seven corresponding positions in total: Left (L), Above (A), AboveRight (AR), LeftBottom (LB), AboveLeft (AL), Colocated (Col) and Colocated-RightBottom (Col-RB).

On each causal Merge block candidate P', we conduct the motion estimation process to find the new MV that minimizes the distortion between the pointed reference block and P'. Instead of searching in the entire frame which is very time consuming, the search range is arbitrarily set and centered around the position pointed by the original MV. Exhaustive search is first tested, then a fast search is also proposed.

In the end of this motion estimation phase, we obtain the MV $\overrightarrow{mv}_{rec}$ that points to a block that minimizes the distortion with P'. This newly computed MV is finally inserted into the Merge list, replacing original MV candidate $\overrightarrow{mv}_{orig}$. This replacement can be performed on only one candidate, or on all five candidates of the Merge list. In other terms, the Merge motion candidates are adjusted by using the proposed motion re-estimation. This replacement allows the motion adjustment without

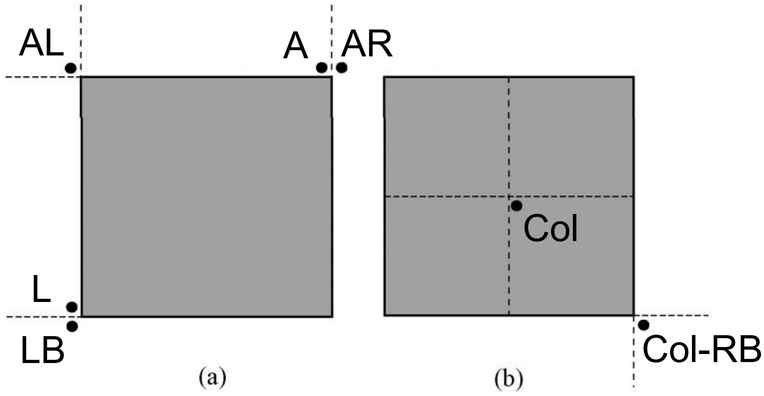


Figure 5.2 – *Merge block candidates covering different positions are used as candidates for the SDec reference (Source: (Helle et al. 2012)).*

modifying the existing HEVC signaling scheme for the Merge candidates.

Since the re-estimated MV computed on each candidate replaces systematically the original Merge MV candidate, we propose to keep the existing HEVC signaling scheme to signal the proposed SDec mode along with the selected candidate. As represented in figure 5.3, the conventional Merge mode which inherits the original MV of a causal block is replaced by the SDec mode which inherits instead the re-estimated MV of the same causal block. The Merge index, which indicates the selected block candidate to inherit the MV, can be assimilated to the syntax element *sdec_ref* that signals the selected candidate to be the SDec reference.

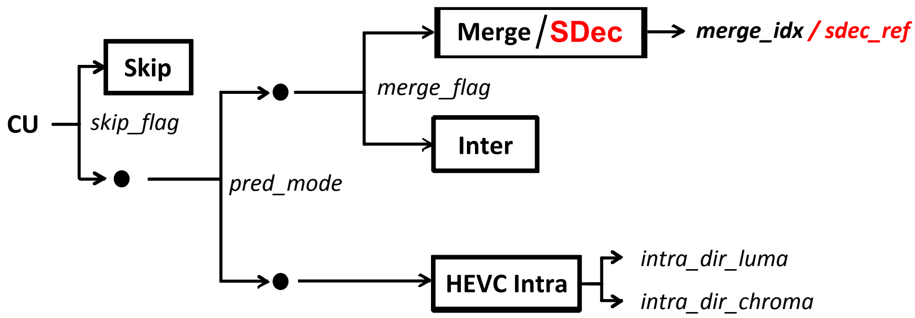


Figure 5.3 – *Signaling scheme of the SDec approach using re-estimated MV.*

5.3.2 Experimental results

In this section, we first propose to evaluate different parameters related to the proposed SDec scheme using re-estimated motion parameters. Experiments refering to the selection of the SDec reference among the Merge block candidates are presented. Then, we evaluate the optimal search range used for the motion re-estimation. Next, we evaluate the best position in the Merge list to insert the SDec reference. Finally, the best configuration is conducted on both standard and extended test sets in order to evaluate the coding performance of the method.

5.3.2.1 Selection of block to be the SDec reference

In this section, we evaluate different block candidates, on which motion parameters are re-estimated, to be the SDec reference. We first conduct experiments with a single candidate. Different cases are tested by using Merge candidates, which can be classified into spatial blocks such as L, A, AR, LB, AL blocks, and temporal blocks such as Col and Col-RB blocks. The case where motion re-estimation is applied for all Merge block candidates is also tested.

The following configuration is used:

Codec: HEVC, test model version 12.0 (HM12)
QPs range: Medium bit rate QP = {22, 27, 32, 37}
Test sequences: CTC test set for HEVC, shortened to 2 seconds
Evaluation: Average B-D rate on all P-frames in LP

Table 5.1 represents the experimental results where different Merge block candidates are used as the SDec reference.

	L	**A**	**AR**	**LB**	**AL**	**Col-RB/Col**	**All cands.**
Class A	0.1	0.0	0.1	0.0	0.0	-0.1	0.1
Class B	0.1	0.0	-0.1	0.0	0.0	-0.2	0.2
Class C	0.1	0.1	0.0	0.0	0.0	-0.2	0.2
Class D	0.2	0.1	0.1	0.1	0.1	-0.1	0.2
Class E	0.6	0.1	0.1	0.1	0.1	0.1	0.9
Class F	-0.6	-0.5	0.2	-0.1	-0.2	-0.5	-0.4
Average	**0.1**	**-0.1**	**0.1**	**0.0**	**0.0**	**-0.2**	**0.2**
Max	-3.5	-2.9	-0.4	-0.3	-0.7	-2.6	-3.6
EncTime	1608%	1012%	253%	148%	241%	5193%	8186%
DecTime	9856%	3460%	729%	234%	450%	1352%	12366%

Table 5.1 – *Coding performance of the SDec scheme inheriting re-estimated MV, with different Merge block candidates as the candidates for the SDec reference block (Ref: HM12).*

According to table 5.1, if the re-estimation process is applied on all Merge candidates, an average loss of 0.2% is observed on the test set. If the scheme is applied only on a single Merge candidate, the use of temporal Merge blocks (derived Col-RB or Col) as the SDec reference provides the best result, with average gain of -0.2% on the entire test set. Gain up to -2.6% is achieved for the sequence "SlideEditing_720p".

However, the encoding time is drastically increased due to the motion re-estimation process conducted on the temporal Merge block Col-RB/Col which is available and inserted in the Merge list much more often than other spatial Merge blocks. This is because of the quad-tree partitioning that sometimes limits the availability of spatial candidates: L block is not available on the left edge of the frame for example, and LB block is even less often available. This problem does not concern the temporal Col-RB/Col block.

Using L block as the SDec reference has the longest decoding time because at the decoder side, L block, being inserted at the first position in the Merge list, is most often selected. There is thus more motion re-

restimation processes conducted by the decoder if we consider L block as the SDec reference.

5.3.2.2 Search range for the block matching based motion re-estimation

For the process of motion re-estimation, exhautive search is initially exploited. With a larger search range, there is more likelihood to find an accurate MV that minimizes the distortion between the pointed reference block and the current block. However, the complexity is more significant. We propose to test different values for the search range: from 256 gradually down to 128, 64, 32, 16, 8 and 4 pixels. The experimental results are provided in table 5.2. Test configuration is similar as in 5.3.2.1, with Col-RB/Col being the single SDec reference on it the motion re-estimation is conducted.

	Full search						
Search range	**256**	**128**	**64**	**32**	**16**	**8**	**4**
Class A	N/A	N/A	-0.1	-0.1	-0.1	-0.1	-0.1
Class B	N/A	N/A	-0.2	-0.2	-0.2	-0.2	-0.1
Class C	-0.1	-0.1	-0.2	-0.2	-0.2	-0.2	-0.1
Class D	0.0	0.0	-0.1	-0.2	-0.1	-0.1	-0.2
Class E	N/A	0.0	0.1	0.1	0.0	0.0	-0.1
Class F	N/A	-0.7	-0.5	0.2	0.1	0.1	0.7
Average	**N/A**	**N/A**	**-0.2**	**-0.1**	**-0.1**	**-0.1**	**0.0**
Max	N/A	N/A	-2.6	-0.5	-0.4	-0.4	-0.4
EncTime	N/A	N/A	5193%	1581%	588%	321%	250%
DecTime	N/A	N/A	1352%	445%	202%	137%	128%

Table 5.2 – *Performance in function of the search range of the SDec scheme inheriting the MV re-estimated using block matching technique - Full search (Ref: HM12).*

We observe that search ranges of 256 and 128 pixels require a very significant runtime (roughly 55 hours to encode one frame of a sequence in class A while the reference HM software takes about 10 minutes), making them inapplicable. For search ranges of 64 pixels and smaller, large search range provides indeed better result than small range. Search range of 64 pixels yields the best result, with -0.2% in average gain and a maximum gain of -2.6%. However, the runtime is also increased, being about three times higher compared to a range of 32 pixels. Reducing the search range shortens indeed the runtime, but also degrades the coding gain.

Aside limiting the search range, we propose to exploit the Test Zone (TZ) fast search algorithm for estimating MVs, available in HM12, in order to reduce the complexity. The algorithm consists of four steps described as follows:

1. Selection of the search center: establish a set of search center candidates which are pointed by the MV obtained from the median prediction, by the MV of the left, the up and the upper right position in the corresponding blocks of the reference frame and by the MV at the position (0,0). Choose the candidate that has the smallest matching error as the search center for step 2.

2. Initial grid search: run the search using diamond or square pattern with different stride lengths ranging from 1 to the predefined search range, in multiples of 2. Find out the smallest matching error point as the search center for step 3.
3. Raster search: perform a full raster search within a limited search range centered around the point computed in step 2.
4. Raster/Star refinement: perform around the optimal point obtained from step 3 the diamond or square search, with stride lengths decreasing according to the exponential of 2. The current starting point location is updated in every pass of the search.

Table 5.3 provides the performance of the proposed fast search, with different search range values.

	TZ fast search								
Search range	**1024**	**512**	**256**	**128**	**64**	**32**	**16**	**8**	**4**
Class A	0.0	-0.1	-0.1	-0.1	-0.1	-0.1	-0.1	-0.1	-0.1
Class B	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2
Class C	-0.1	-0.1	-0.1	-0.2	-0.1	-0.2	-0.1	-0.1	-0.1
Class D	-0.2	-0.2	-0.1	-0.1	-0.1	0.0	-0.1	-0.1	-0.1
Class E	-0.1	0.0	-0.1	0.0	0.0	0.0	-0.1	0.0	0.1
Class F	-1.4	-1.5	-1.4	-0.6	-1.0	0.1	-0.2	-0.3	0.3
Average	**-0.3**	**-0.3**	**-0.3**	**-0.2**	**-0.2**	**-0.1**	**-0.1**	**-0.1**	**0.0**
Max	-5.0	-5.2	-5.4	-1.8	-3.3	-0.5	-0.4	-0.6	-0.8
EncTime	338%	285%	245%	230%	232%	213%	212%	211%	210%
DecTime	152%	144%	132%	125%	118%	118%	118%	125%	127%

Table 5.3 – *Performance in function of search range of the SDec scheme inheriting MV re-estimated using block matching technique - TZ fast search (Ref: HM12).*

According to table 5.3, the proposed TZ fast search is very efficient, reducing significantly the complexity while preserving the gain in average. Indeed, using TZ fast search with search range of 64 pixels for example also yields an average gain of -0.2% similar as when using the full search, but the encoding time is only 232% instead of 5193% compared to the use of the full search. It is also observed that large search range provides in general better result than small range with longer runtime, similarly as the previous experiment with the exhaustive search. The search range of 256 pixels gives the best compromise between coding gain and runtime, yielding -0.3% in average gain and a maximum gain of -5.4%. The encoding and decoding time are respectively 245% and 132%.

5.3.2.3 Position of the SDec reference in the Merge list

The Merge list is composed of at most four spatial candidates selected among L, A, AR, LB, AL blocks and in that priority order, followed by a temporal candidate selected among Col-RB and Col blocks. In this test where Col-RB/Col is taken as the SDec reference so that its motion parameters are re-estimated, we propose to evaluate the compression gain of the scheme in function of the order of the SDec reference in the Merge list. More precisely, we insert Col-RB/Col in different positions between spatial candidates L, A, AR, LB, AL as shown in table 5.4 and conduct the SDec scheme.

	Gain	Max.
1st position	0.6%	-0.8%
After L	0.1%	-4.9%
After A	-0.3%	-4.7%
After AR	-0.3%	-4.6%
After LB	-0.3%	-5.5%
After AL	-0.3%	-5.4%

Table 5.4 – *Gain of the SDec scheme inheriting re-estimated MV in function of the position of the SDec reference Col-RB/Col in the Merge list (Ref: HM12).*

Experimentals results shows that when taking Col-RB/Col as the SDec reference which has its motion parameters re-estimated, inserting Col-RB/Col just after the spatial candidate LB yields the best coding performance, with -0.3% in average gain and -5.5% in maximum gain.

5.3.2.4 Best configuration

After conducting previous tests concerning different parameters related to the proposed method, we choose following configuration that provides the best compromise between coding gain and complexity: temporal Merge candidate Col-RB/Col is taken as the SDec reference and its corresponding MV is systematically replaced by the new MV that is re-estimated; TZ fast search with search range of 256 pixels is used for the re-estimation process; the candidate Col-RB/Col is inserted just after the spatial candidate LB in the Merge list instead of at the last position as in the reference HM12. We give the coding results in LP configuration, on the standard CTC test set and also on an extended test set which includes additional sequences. Different coding rates are also used to evaluate the performance of the proposed SDec scheme: low bit rate (LBR) QP 27-32-37-42, medium bit rate (MBR) QP 22-27-32-37 and high bit rate QP 27-32-37-42.

The following configuration is used:

Codec: HEVC, test model version 12.0 (HM12) QPs range: Low bit rate QP = {27, 32, 37, 42}, Medium bit rate QP = {22, 27, 32, 37}, High bit rate QP = {17, 22, 27, 32} Test sequences: CTC test set for HEVC and additional sequences Evaluation: Average B-D rate on all frames in LP

On the HEVC test set, we observe a systematic gain, although not significant, on every test sequence classes. Average gains of -0.1%, -0.2% and -0.1% are obtained on the HEVC test set respectively in LBR, MBR and HBR. The runtime at both encoder and decoder sides is increased respectively of 245% and 132%.

On the extended test set with additional sequences, it is observed that proposed method is fairly efficient for sequences containing lot of small textures or complex motion difficult to be predicted by conventional motion estimation (e.g. sequence "RollingTomatoes_1080p" illustrated in fig-

Sequences class	*LP*		
	LBR	**MBR**	**HBR**
Class A	-0.1	-0.1	-0.1
Class B	-0.1	-0.1	-0.1
Class C	-0.1	-0.1	-0.1
Class D	-0.1	-0.1	-0.1
Class E	-0.1	-0.1	-0.1
Class F	-0.4	-0.3	-0.3
Average	**-0.1**	**-0.2**	**-0.1**
Max gain	-0.9	-0.9	-0.8
HomelessSleeping 2160p	-0.3	-0.3	0.0
RollingTomatoes 1080p	-0.5	-0.6	-0.2
S8FootFranceRou 1080p	-0.2	-0.2	-0.1
S18DavisCup 1080p	-0.3	-0.3	-0.2
ToysAndCalendar 1080p	-0.3	-0.2	-0.1
WalkingCouple 1080p	-0.2	-0.1	-0.1
Average	**-0.3**	**-0.3**	**-0.1**
Enc Time	245%		
Dec Time	132%		

Table 5.5 – *Bit rate savings (%) of the SDec scheme inheriting MV re-estimated using block matching technique (Ref: HM2)*

ure 5.4). A motion re-estimation is thus useful in those cases to improve the precision of inherited motion parameters. We also remark that the proposed method does not perform well on sequences with high frame rate. Indeed, in this type of sequences, motion parameters computed by conventional methods are often already accurate because there is more correlation between frames to be exploited, making the motion re-estimation process useless.

Figure 5.4 – *Sequence "RollingTomatoes_1080p" containing motion difficult to be predicted.*

5.3.3 Statistical analysis

We provide some statistics by evaluating how many MVs are effectively modified by the proposed method of MV re-estimation. The result, given in table 5.6, shows that on the HEVC test set, an average of 38.5% of total re-estimated MVs differ from original MVs when taking the Merge candidate Col-RB/Col as the single SDec reference. This significant ratio proves that the proposed method has a large impact on the Merge candidates, by often replacing their motion parameters by the re-estimated ones.

QP	Percentage
22	43.7%
27	40.4%
32	37.9%
37	31.9%
Avg.	**38.5%**

Table 5.6 – *Percentage of re-estimated MVs that are different than original MVs - SDec scheme inheriting re-estimated MV with Col-RB/Col as SDec reference.*

5.4 Implementation with motion parameters re-estimated using Optical Flow technique

In this section, we propose to exploit Optical Flow (OF) instead of block matching technique for the motion re-estimation process on the SDec reference. Unlike block matching based motion estimation which produces a single MV for a given block, OF technique provides a dense motion vector field (DMVF) where each pixel is associated to a MV. The granularity is thus on a finer level. Therefore, the re-estimated motion parameter is improved by using OF technique compared to the previous use of block matching. In the proposed approach, DMVFs are computed using the OF algorithm which is found in (Liu 2009) and is based on two already reconstructed frames. It is important to note that, similarly to the MV re-estimated using block matching, the DMVF is not signaled in the bit stream because the decoder is able to retrieve it by conducting the motion re-estimation on the SDec reference, saving thus the signaling cost for the DMVF.

5.4.1 Application in 2D coding

5.4.1.1 Description

As illustrated in figure 5.5, let us suppose that we are encoding a current block P in the frame at instant t_i. A block P' located in the reconstructed frame at instant t_{i-1} is taken as the SDec reference. We suppose that P' has a MV denoted by $\overrightarrow{mv}_{orig}$. Unlike the block matching based motion re-estimation previously presented that provides $\overrightarrow{mv}_{rec}$, optical flow technique is exploited to compute the DMVF containing MVs for each pixel of P' from instant t_{i-1} to t_{i-2}. This DMVF is then considered as the re-estimated motion parameters of P' that will be inherited to encode the current block P.

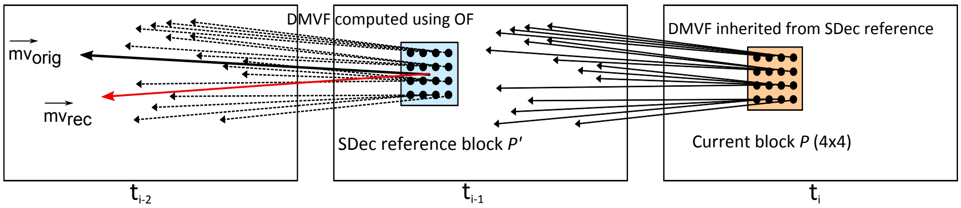


Figure 5.5 – *SDec scheme inheriting dense MVs field computed using OF technique.*

The computation of the optical flow is performed at the frame level, i.e. the algorithm requires entire reconstructed frames at both instants t_{i-1} and t_{i-2} as inputs and provides at the output, for every pixel in the frame at instant t_{i-1}, an MV that points to frame at instant t_{i-2}.

As in previous section, we propose to exploit Merge block candidates to be the SDec reference. Since the proposed OF based motion re-estimation conducted on the SDec reference involves the computation of the DMVF and requires the entire reconstructed frame, it is not possible to use spatial Merge candidates located in the current frame as the SDec reference. This limitation is purely due to the algorithm being used to compute the OF. We consider therefore following temporal Merge block candidates in previously decoded frame as the single SDec reference for our tests:

– Col block,
– Col-RB block.

The signaling scheme for blocks encoded in SDec mode remains the same as in previous section using block matching based motion re-estimation and is unchanged compared to the HEVC signaling scheme: since the re-estimated motion parameters replaces systematically the Merge motion candidate being used as the SDec reference, the SDec mode is signaled by its corresponding Merge index.

5.4.1.2 Experimental results

Among Col and Col-RB blocks, we will evaluate which block is more suitable to be the SDec reference. The following configuration is used for the tests:

Codec: HEVC, test model version 12.0 (HM12)
QPs range: Medium bit rate QP = {22, 27, 32, 37}
Test sequences: CTC test set for HEVC, shortened to 2 seconds
Evaluation: Average B-D rate on all P-frames in LP

According to table 5.7, We observe that using Col block as the SDec reference yields better gain in average than using Col-RB block, with -0.2% of average gain on the HEVC test set. The proposed method performs particularly well for videoconferencing sequences (class E) which contain little or no movement, with -0.9% of gain in average. The maximum gain of -1.1% is also obtained for the videoconferencing sequence "KristenAnd-Sara_720p" illustrated in figure 5.6. Indeed, since there is little motion

	Col	**Col-RB**
Class A	-0.1	0.1
Class B	-0.2	-0.1
Class C	-0.2	0.0
Class D	-0.3	-0.1
Class E	-0.9	-0.1
Class F	0.2	0.1
Average	**-0.2**	**0.0**
Max	-1.1	-0.6
EncTime	210%	218%
DecTime	39338%	40415%

Table 5.7 – *Compression performance of the SDec scheme which inherits the DMVF computed using OF technique, with different temporal Merge candidates as the SDec reference (Ref: HM12).*

variation in time, the DMVF computed on a block at instant t_{i-1} is more likely to be adapted to encode the current block at instant t_i.

Figure 5.6 – *Videoconferencing sequence "KristenAndSara_720p" containing little motion variation.*

We propose to further investigate this configuration of using Col block as the SDec reference by conducting extended test on different QPs ranges and additional sequences as follows:

Codec: HEVC, test model version 12.0 (HM12) QPs range: Low bit rate QP = {27, 32, 37, 42}, Medium bit rate QP = {22, 27, 32, 37}, High bit rate QP = {17, 22, 27, 32} Test sequences: CTC test set for HEVC and additional sequences Evaluation: Average B-D rate on all frames in LP

According to the table 5.8, on the HEVC test set, the proposed method yields average gains of -0.3%, -0.2% and -0.1% respectively under LBR, MBR and HBR encoding rates. The performance in LBR can be explained by the underperformance of conventional block matching based motion estimation due to the limited quality of reconstructed frames. The method performs indeed very well on videoconferencing sequences, achieving -1.0% for sequence "FourPeople_720p". We finally remark that

Sequences class	*LP*		
	LBR	**MBR**	**HBR**
Class A	-0.2	-0.1	-0.1
Class B	-0.3	-0.2	-0.1
Class C	-0.3	-0.2	-0.1
Class D	-0.3	-0.2	-0.2
Class E	-0.9	-0.7	-0.5
Class F	0.0	0.1	0.0
Average	**-0.3**	**-0.2**	**-0.1**
Max gain	-1.3	-1.0	-0.7
BonneFranquette 2160p	-0.8	-0.6	-0.3
BalletExt2 1080p	-0.7	-0.4	-0.1
BasketBall 1080p	-0.6	-0.3	-0.1
Ccett 1080p	-0.4	-0.3	-0.1
MvAvatar 1080p	-0.3	-0.2	-0.1
WalkingCouple 1080p	-0.1	-0.1	-0.8
Average	**-0.5**	**-0.3**	**-0.3**
Enc Time	212%		
Dec Time	39338%		

Table 5.8 – *Performance of the SDec scheme inheriting motion parameters re-estimated using OF technique (Ref: HM12).*

sequences with high frame rate also yield interesting gains, for example "BalletExt2_1080p 120Hz" or "Ccett_1080p 100Hz". Indeed, since the texture variation is not significant from a frame taken at instant t_i to the next frame at instant t_{i+1}, the effect of the proposed method on high frame rate sequences can be considered similar as on videoconferencing sequences.

Table 5.9 provides the distribution of the Col block, which is selected as the SDec reference, among all Merge candidates for both approaches using respectively the block matching based and using the OF based motion re-estimation.

QP	**Block matching based**	**OF based**
22	2.1%	2.1%
27	2.3%	2.7%
32	2.5%	3.0%
37	2.5%	2.8%
Average	**2.4%**	**2.7%**

Table 5.9 – *Percentage distribution of Col Merge block candidate among all Merge candidates in approaches using block matching based and OF based motion re-estimation.*

We observe that Col Merge candidate is more likely to be selected when using OF based re-estimation compared to the version using block matching, proving that the OF based re-estimated motion parameter is more accurate than its block matching based counterpart.

5.4.2 Application in 3D coding

The motivation to apply the proposed SDec scheme in 3D video coding is that there is strong correlation between different views. For a current block in a dependent view, its corresponding block pointed by the disparity vector (DV) and located in the base view is very likely to be correlated with similar motion parameter. Therefore, by inheriting re-estimated motion parameters of the reference block in the base view to encode the current block in a dependent view without the need to signal those parameters in the bit stream, improvement in coding performance can be obtained.

5.4.2.1 Description

The basic idea is to improve dependent views by inheriting motion parameters from the base view. The efficiency of OF is exploited to re-estimate the motion parameters of the SDec reference located in the base view, providing a dense motion vectors field (DMVF) which maps a MV to each pixel of the block. This DMVF is finally inherited to encode the current block located in a dependent view.

More precisely, let us suppose that the current block P is in a dependent view at instant t_i as presented in figure 5.7. Let P' be the SDec reference block that is located in the base view and pointed by the DV. The OF based motion re-estimation process provides for P' a DMVF that is considered as the re-estimated motion parameter of P' and that consists of MVs that map each pixel of P' to a pixel in the previously reconstructed frame at instant t_{i-1} in the base view. This DMVF is finally inherited to encode P in dependent view without being signaled in the bit stream.

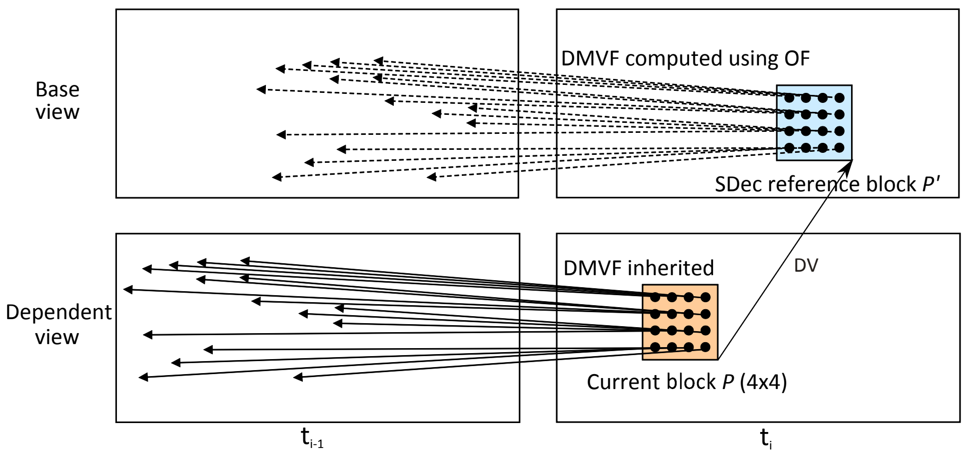


Figure 5.7 – *3D SDec scheme inheriting dense MVs field computed using Optical Flow technique.*

We remark that if the block encoded in SDec mode is signaled as the first candidate inserted in the 3D-HEVC Merge list, the whole scheme becomes the Warped Optical Flow (WOF) approach originally proposed in (Mora 2014). For this reason, we propose first to implement WOF technique in the 3D-HEVC software test model version 9.3 (HTM9.3) (Tech 2012). It will turn out later that the implementation does not provide good performance due to the competition with Sub-PU Inter-View Motion Prediction (SP-IVMP), a technique adopted later into HTM. This observation motivates us to propose an improvement where WOF is applied on a sub-PU level (instead of PU level) and where WOF and SP-IVMP techniques are put in a SDec competition which allows to save the signaling overhead dedicated for the index of the selected candidate. Both WOF and SP-IVMP are already presented in the chapter 1 referring to the state of the art.

5.4.2.2 Implementation of WOF

In WOF technique, the current PU is encoded by inheriting the DMVF from the base view and is signaled as a Merge candidate. The signaling of PUs encoded with WOF involves the creation of a modified Merge list which is established by following steps:

- First, the list of HEVC Merge candidates is computed, using the default 3D-HEVC process.
- Then, the motion parameter of the PU encoded with WOF is inserted as a Merge candidate in the first position of the Merge list.
- Finally, the remaining inter-view candidates (including SP-IVMP candidate) are inserted, using the default 3D-HEVC process. The maximum number of 6 candidates in 3D-HEVC is preserved.

We implement this WOF method in the HTM9.3 software. The introduced WOF candidate is in competition with the SP-IVMP candidate since both of them are inserted in the Merge list. In order to evaluate the coding performance of each candidate separately, we provide following tests using different versions of HTM9.3 where SP-IVMP and WOF are activated/deactivated:

- Test 1: Evaluate the SP-IVMP candidate by comparing to a reference HTM9.3 that deactivates SP-IVMP
- Test 2: Evaluate the WOF candidate by comparing to a reference HTM9.3 that deactivates SP-IVMP
- Test 3: Evaluate the combined use of SP-IVMP and WOF candidates by comparing to a reference HTM9.3 that deactivates SP-IVMP
- Test 4: Evaluate the combined use of SP-IVMP and WOF candidates by comparing to the reference HTM9.3 (that has SP-IVMP activated by default)

Experiments are conducted following common test conditions (CTCs) defined by JCT-3V (Rusanovsky et al.).

Test configuration can be summarized as follows:

Codec: 3D-HEVC, test model version 9.3 (HTM9.3) QPs range: Medium bit rate with four QP combinations for texture and depth: (25;34), (30;39), (35;42) and (40;45). Test sequences: CTC test set for 3D-HEVC, shortened to 1 second Evaluation: Average B-D rate on all frames

Table 5.10 summarizes the average gain results of all four tests. The "Video" column shows the gains on the two side dependent views (1 and 2) and in average on all coded views texture. Column "Video 0" is omitted since the approach is not applied to base view coding, thus there is no

gain impact. The "Video total" column gives results on all coded views. Result in the "Synt." column provides gain on all synthesized views.

Sub-PU size	Video			Video total	Synt.	Ratio runtime	
	1	2	Avg.			Enc	Dec
Test 1	-5.3	-5.3	-1.2	-1.1	-0.8	110%	110%
Test 2	-4.5	-4.2	-1.4	-1.4	-1.3	130%	16000%
Test 3	-5.4	-5.3	-1.7	-1.5	-1.4	130%	16300%
Test 4	-0.1	0.0	-0.5	-0.4	-0.6	120%	15000%

Table 5.10 – *Summary of tests related to the evaluation of SP-IVMP and WOF techniques. Compression gain is obtained on the standard 3D-HEVC test set.*

According to table 5.10, we observe that both SP-IVMP and WOF techniques, if used separately, perform very well with significant gain in average (test 1 and test 2). However, the runtime of WOF technique is also increased significantly due to the OF computation performed on the entire frame level.

The combined use of SP-IVMP and WOF candidates yields even further gain (test 3). WOF candidate provides indeed an improvement to the coding performance. However, the increase in gain obtained by using additionally WOF is not as significant as the gain obtained using WOF separately.

Compared to the reference HTM9.3 where SP-IVMP is activated by default, this combined use of SP-IVMP and WOF candidates does provide some coding gain (test 4), but not as significant as in other tested cases. This observation confirms that WOF candidate encounters indeed a rude concurrence from the SP-IVMP candidate. Indeed, the coding gains provided by WOF candidate and by SP-IVMP candidate are in the same nature: both of them are based on techniques that inherit motion parameters from the base view; and the prediction accuracy of both techniques is improved by using finer granularity for the motion estimation (i.e. on pixel level and on sub-PU level respectively).

5.4.2.3 Competition between WOF and SP-IVMP

As both WOF and SP-IVMP techniques share some common points as previously mentioned, instead of introducing WOF as a new Merge candidate on the PU level as in previous section, we propose to:

- Introduce WOF candidate on sub-PU level,
- And put both the WOF and SP-IVMP candidates in a SDec competition to save the signaling cost of the selected candidate.

More precisely, let us assume that we are encoding a PU P in a dependent view at instant t_i as illustrated in figure 5.8. Let SP_i and $refSP_i$ be respectively the sub-PUs inside P and their corresponding reference sub-PUs in the base view determined using a DV. For a sub-PU SP_i, its reference $refSP_i$ will have a DMVF $\overrightarrow{dmvf_i}$ computed by OF technique based on reconstructed frames between instant t_{i-1} and t_i in the base view, and a MV $\overrightarrow{mv_i}$ if coded in MCP. Both the DMVF and the MV point to the previous frame at instant t_{i-1} in the base view already reconstructed. The decoder also has access to these frames, thus it can perform similar

OF computation to obtain $\overrightarrow{dmvf_i}$, removing the need to transmit all the DMVFs corresponding to the sub-PUs. Let $refSP_i^{\overrightarrow{mv}}$ and $refSP_i^{\overrightarrow{dmvf}}$ be the motion compensated sub-PU of $refSP_i$ respectively using the MV $\overrightarrow{mv_i}$ and the DMVF $\overrightarrow{dmvf_i}$, both found in the reconstructed frame at instant t_{i-1}.

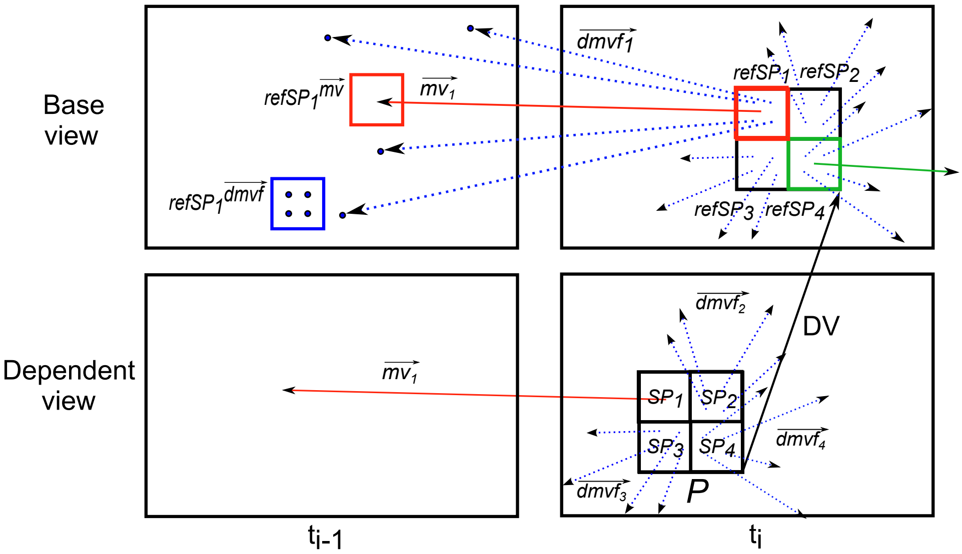


Figure 5.8 – *DMVF and regular MV candidates are in competition on each reference sub-PU $refSP_i$ in base view. The optimal candidate computed on $refSP_i$ is inherited for SP_i in dependent view.*

The SDec scheme that conducts the competition between both SP-IVMP based and WOF based candidates for each sub-PU SP_i can be summarized in three following steps:

– ***Step 1***: Determine $refSP_i$ as the SDec reference of SP_i using the DV.

– ***Step 2***: Conduct the competition on the SDec reference $refSP_i$ between $\overrightarrow{mv_i}$ and the DMVF $\overrightarrow{dmvf_i}$ using SAD as choice criterion.

  More precisely, for each SP_i, we propose to conduct a SDec competition between the SP-IVMP based MV $\overrightarrow{mv_i}$ and the WOF based DMVF $\overrightarrow{dmvf_i}$ on the SDec reference $refSP_i$. We simply use the sum of absolute difference (SAD) as the distortion metric to evaluate which candidate predicts better the $refSP_i$, resulting in comparing the SAD values of $refSP_i^{\overrightarrow{mv}}$ and of $refSP_i^{\overrightarrow{dmvf}}$ regarding $refSP_i$. The optimal candidate resulting from this competition on the SDec reference $refSP_i$ in the base view is finally inherited to be used as motion parameter for SP_i in the dependent view and not signaled in the bit stream.

  Note that the evaluation of optimal candidate in the SDec scheme, although performed on $refSP_i$ instead of on current SP_i as in conventional scheme, is still relevant: since SP_i in a dependent view and its corresponding reference $refSP_i$ in the base view are correlated due to the inter-view correlation, the chosen candidate which is optimal regarding to $refSP_i$ will probably be optimal for SP_i.

– ***Step 3***: Derive the motion parameter for SP_i by inheriting the optimal candidate computed on $refSP_i$

During the decoding process, this SDec competition on the reference sub-PUs is performed by the decoder exactly the same way as by the encoder. Therefore, the optimal candidate derived for each sub-PU can be correctly retrieved by the SDec decoder without being signaled in the bit stream, guarantying the decodability for encoded sequences. We also note that for a SP_i, if $\overrightarrow{mv_i}$ is not available because the reference $refSP_i$ is not encoded in MCP, the DMVF $\overrightarrow{dmvf_i}$ computed by OF technique is automatically selected.

The compression performance of our approach comes from the use of the WOF based DMVF candidate which allows an efficient prediction without suffering the burden of signaling overhead. Moreover, by using the SDec scheme to conduct the competition on the sub-PU level, a large portion of signaling overhead dedicated to transmit the index of the selected candidate for every sub-PU in P is removed, thus further contributing to the overall performance. Indeed, no additional signaling is performed in the proposed method, neither for the DMVF nor for the index of the selected competing candidate. Without the SDec scheme, a significant amount of bits is required to signal the selected candidate for every sub-PU, thus reducing severely the performance.

The proposed method is implemented in HTM9.3. The programming code used for OF computation comes from a package downloadable from (Liu 2012) and is integrated in HTM9.3. Our approach is not applied on depth layer. Moreover, it is deactivated when the weighting factor of Advanced Residual Prediction (ARP) is equal to one, which corresponds to the case where existing SP-IVMP MV candidate displays a good performance in exploiting spatio-temporal correlation. Therefore, the proposed DMVF candidate is not introduced for competition in this case.

Experiments are conducted following CTCs for 3D coding. Four QP combinations for texture and depth were considered: (25;34), (30;39), (35;42) and (40;45). The test set contains eight sequences defined in CTCs. All of them are grouped in two classes depending on their resolution (1024×768 and 1920×1088). Each sequence is composed of three texture and three depth views corresponding to one central base view and two side dependent views. After the encoding, three intermediate views are synthesized between the left and the center view, and another three between the center and the right views. PSNR on synthesized views are computed with respect to synthesized views that are rendered with uncompressed original texture and depth views. All coding gains are measured using B-D rate.

Following test configuration is used:

Codec: 3D-HEVC, test model version 9.3 (HTM9.3) QPs range: Medium bit rate with four QP combinations for texture and depth: (25;34), (30;39), (35;42) and (40;45). Test sequences: CTC test set for 3D-HEVC Evaluation: Average B-D rate on all frames

Table 5.11 gives the bit rate savings of proposed method compared to the default HTM9.3.

Sequences	**Video**			**Video total**	**Synt.**
	1	**2**	**Avg.**		
Balloons	-3.4	-4.0	-1.4	-1.3	-1.3
Kendo	-3.5	-3.7	-1.3	-1.2	-1.2
Newspaper	-1.8	-1.6	-0.6	-0.5	-0.5
GT Fly	-3.2	-2.7	-0.6	-0.6	-0.5
Poznan Hall2	-2.1	-2.3	-0.8	-0.8	-0.7
Poznan Street	-1.6	-1.9	-0.5	-0.4	-0.5
Dancer	-2.2	-2.0	-0.5	-0.4	-0.5
Shark	-1.6	-1.9	-0.4	-0.3	-0.3
1024×768	-2.9	-3.1	-1.1	-1.0	-1.0
1920×1088	-2.1	-2.1	-0.5	-0.5	-0.5
Average	**-2.4**	**-2.5**	**-0.8**	**-0.7**	**-0.7**
DMVF cand. only	-0.5	-0.1	-0.8	-0.9	-1.1
No SDec scheme	459.2	440.0	119.1	106.8	105.5
EncTime	173%				
DecTime	12830%				

Table 5.11 – *Performance of the method exploiting the competition between WOF and SP-IVMP techniques using SDec scheme (Ref: HTM9.3), showing very good compression ratio. Severe loss is observed if the SDec scheme is not used.*

The experimental results shows that significant coding gains of 2.4% and 2.5% in average are achieved for dependent texture views, confirming the efficiency of OF in providing a better prediction. Gain up to 4.0% is reported for the sequence "Balloons". For both coded and synthesized views, an average gain of 0.7% is observed. Furthermore, is it observed that the gain is systematic for every test sequence, proving that the proposed scheme is very performant in terms of compression capacity.

Encoding time and decoding time are reported as a side information, given that the downloaded OF code integrated in our test is not optimized. Reported runtime are increased by factors of 1.7 and 128 respectively, due to the fact that not only the integrated OF code is not homogenous with the HTM code, but especially because the computation of OF is performed on the entire frame.

Statistically, as shown in table 5.12, the WOF based candidate is selected over the SP-IVMP based candidate in 8.4% of time in average during SDec competitions on reference sub-PUs in the base view. Given the significant gain obtained, this rather low selection rate proves again that OF technique is very efficient and can have significant impact on the compression performance.

Sequences	% WOF based cand.
Balloons	12.6
Kendo	12.6
Newspaper	8.9
GT Fly	2.3
Poznan Hall2	2.4
Poznan Street	6.9
Dancer	7.1
Shark	6.9
1024x768	11.4
1920x1088	5.1
Average	**6.9**

Table 5.12 – *Percentage of WOF based candidate when being in competition with SP-IVMP based candidate.*

For comparison purpose, we also give in the bottom of table 5.11 the results of some tested variants which have roughly similar runtime since they are all based on the exploitation of OF:

– DMVF cand. only: if we only consider the WOF based DMVF candidate and remove the existing SP-IVMP based MV candidate out of the competition, only slight gain is observed on dependent views (i.e. 0.5% and 0.1%).
– No SDec scheme: if the SDec scheme is not used, the evaluation of optimal candidate is performed on the current sub-PU and an index must be transmitted in the bit stream for every sub-PU, resulting in severe loss in coding performance.

Other attempts to improve the performance of proposed method are also tested, but none of them gives an improvement in coding gain:

• Use SSE metric instead of SAD metric for blocks comparison in the SDec process: we replace SAD metric by SSE metric in the second step of the SDec process.

• Use different evaluating conditions to select between WOF based and SP-IVMP based candidates: inter-views correlation at instant t_{i-1} is additionally exploited in the evaluation of both temporal WOF based and SP-IVMP based candidates.

• Further reduce the granularity of SP-IVMP based candidate: we propose to add 4×4 SP-IVMP based candidate in competition with both 8×8 SP-IVMP based and WOF based candidates. There is thus three candidates competing with each other on the reference sub-PU $refSP_i$.

5.4.3 Perspectives

Exploiting OF technique in video coding provides indeed interesting improvements in performance. The major concern is the significant runtime of related methods due to the computation of OF on frame level. Several perspectives can be considered to reduce this runtime.

The runtime could be greatly improved by using an on-the-fly OF computation on PU level and uniquely when it is needed. Indeed, in the proposed 3D coding method that exploits the SDec competition of both SP-IVMP and WOF techniques for example, only 38% of PUs are concerned by the method. There is thus no need to compute the OF for the remaining PUs, which could reduce greatly the runtime.

Another perspective is to optimize the code dedicated to the computation of OF. Large number of parameters are involved in the OF computation. Optimizing those parameters could indeed provide improvements on both coding efficiency and complexity.

Conclusion

In this chapter, we presented several practical applications of the SDec scheme inheriting re-estimated motion parameters of the SDec reference to encode the current block. It is proposed that the motion re-estimation process can be based on block matching technique or OF technique.

By exploiting the block matching based motion re-estimation on a Merge block candidate considered as the SDec reference, systematic gain is obtained on the set of tested sequences. The gain in average is however not significant.

OF based motion re-estimation is proposed for both 2D and 3D coding. Dense MVs field computed by OF technique on the SDec reference is inherited to encode the current block. In 2D, with temporal Merge candidate as the SDec reference, interesting gain is achieved for visioconferencing sequences, reaching -0.7% in average for class E of standard HEVC test set. In 3D, we proposed to introduce both WOF and SP-IVMP techniques in a SDec competition on the sub-PU level, allowing to save the signaling of not only the DMVF but also the index of competing candidates in the bit stream. Significant bit rate reductions of 2.4%and 2.5%for two dependent views, and 0.7% for coded and synthesized views are reported. Runtime is also significantly increased due especially to the computation of OF conducted on the frame level.

As future works, several approaches to reduce the runtime can be considered for the approaches that exploit OF technique. Optimizing the code dedicated to the computation of OF and reducing the scale on which OF is generated are interesting perspectives.

6 SDec based coding scheme combining a multitude of coding modes

In this chapter, a multitude of different coding modes are used during the SDec process. Coding modes that are already exploited in previous chapters, i.e. Intra mode, Intra 1D mode and re-estimated motion, are put in the competition on the SDec reference. First, a description of the proposed scheme is described for 2D coding. Experimental results along with statistical analysis are then presented. After that, the 3D SDec scheme which exploits the three mentioned coding modes is proposed. Experimental results and statistical analysis are also given. Finally, an improvement is proposed, giving the idea to exploit learning techniques to provide an adaptive selection of the SDec reference and the coding modes being used during the SDec process depending on the visual content of encoded sequences.

6.1 SDec scheme exploiting several coding modes applied in 2D

6.1.1 Description

Let us suppose that we are encoding the current block P in the frame at instant t_i. According to the principle of the SDec scheme, a causal block P' found in a reconstructed frame at instant t_j is taken as the SDec reference. Unlike chapters where coding modes are separately exploited for the SDec process (i.e. Intra mode in chapter 3, Intra 1D mode in chapter 4 and Inter with re-estimated motion in chapter 5), all the mentioned coding modes with their parameters are used to encode the SDec reference. The optimal coding mode minimizing the cost is inherited to encode the current block P.

Following figure illustrates the proposed SDec scheme:

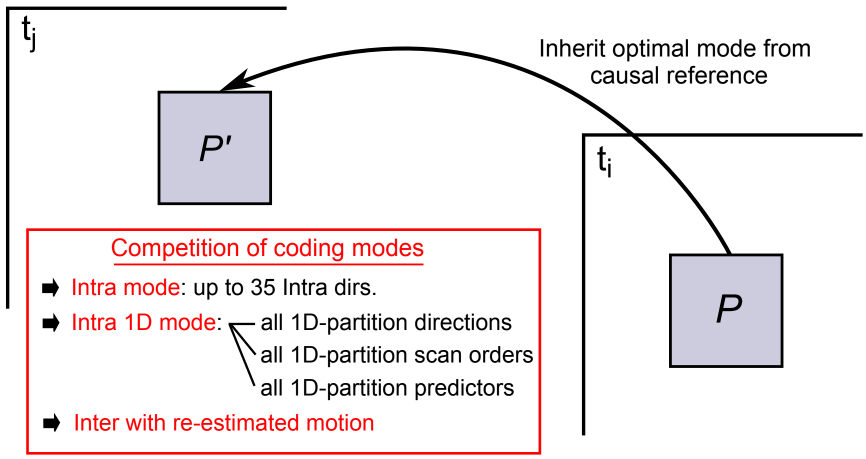


Figure 6.1 – *SDec scheme with competition of several coding modes: Intra, Intra 1D and Inter with re-estimated motion.*

6.1.2 Experimental results

For the configuration of the experiments, we consider only a single candidate to be the SDec reference for simplification purpose: Colocated block for P-,B- frames and Above block for I-frames. The configuration specific for each coding mode exploited in the SDec scheme is based from the optimal configuration related to each coding mode separately and found in previous chapters:

- Intra mode: using an adaptive fast search pre-selection that limits to 8 Intra directions among all 35 available directions.
- Intra 1D mode: only 11 Intra directions (Planar, DC, 2, 6, 10, 14, 18, 22, 26, 30, 34) are exploited.
- Re-estimated motion: Optical Flow technique is exploited to re-estimate the motion of the SDec reference.

Concerning the signaling scheme, the syntax element *sdec_flag* is signaled as illustrated in figure 6.2 using CABAC with three contexts based on neighboring blocks. SDec mode is on par with HEVC Intra mode because, according to a preliminary non-decodable test that evaluates on P-frames the percentage of total blocks encoded in each coding mode without accouting its signaling cost, SDec mode is used on nearly 10% and can compete with HEVC Intra mode (6%). Putting *sdec_flag* in other position of the signaling tree will heavily penalize other coding modes (for example Skip or Merge with respectively about 50% and 20% of total encoded blocks), resulting in consequence a loss in compression performance. The syntax element *sdec_ref* is not signaled because a single candidate is used for the SDec reference in this proposed practical application.

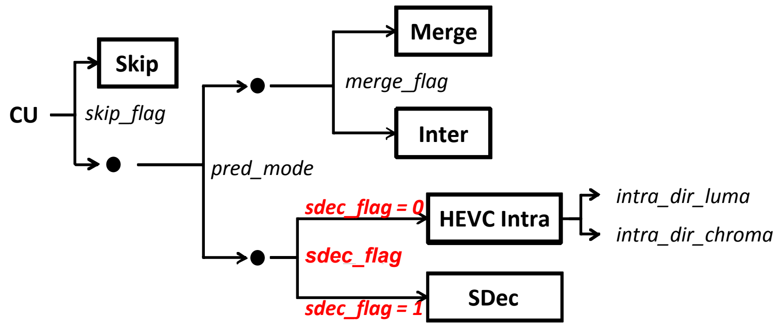


Figure 6.2 – *Signaling scheme for the SDec mode in the proposed practical application.*

We conduct several tests where different combinations of coding modes (among Intra mode, Intra 1D mode and re-estimated motion) are used during the SDec process. The percentage of each coding mode selected as optimal on the SDec reference and thus inherited for the current block is also given.

Following test environment is used for all experiments:

Codec: HEVC, test model version 12.0 (HM12) QPs range: Medium bit rate QP = {22, 27, 32, 37} Test sequences: CTC test set for HEVC, shortened to 5 frames in AI and to 2 seconds in RA and LP Evaluation: Average B-D rate on all frames in AI and RA, and on P-frames in LP

Table 6.1 presents the experimental results, with each coding modes (Intra, Intra 1D and re-estimated motion) being exploited separately then in combination with the others during the SDec process.

We observe that combining Intra and Intra 1D modes does not yield improvement in average gain over the use of only Intra mode, with -0.2% of average gain in both AI and RA configurations compared to -0.3% when only Intra mode is used. The maximum gain is however increased in RA and LP configurations. This result can be explained by the limited number of test sequences, among the standard HEVC test set, that have the content suitable for Intra 1D mode: Intra 1D yields significant gain only on sequences in class F, particularly the sequence "SlideEditing_1280×720" containing complex texture. Indeed, for this type of sequences, additionally exploiting Intra 1D mode allows to further increase the coding gain. Results on extra test sequences, given in the next experiment that is conducted on a full extended test set, will confirm this observation.

If we combine additionally the re-estimated motion along with Intra and Intra 1D modes, the coding performance in LP is improved systematically and yields an average gain of -0.8%, proving that the proposed SDec

		Avg. gain	Max.	EncTime	DecTime
Intra	AI	-0.3	-0.8	285%	383%
	RA	-0.3	-0.7	185%	244%
	LP	-0.7	-3.9	183%	244%
Intra 1D	AI	-0.1	-0.4	397%	232%
	RA	0.0	-0.2	232%	211%
	LP	-0.3	-3.0	231%	190%
Re-estimated motion	AI				
	RA	0.0	-0.2	216%	29739%
	LP	-0.2	-1.4	320%	39286%
Intra & Intra 1D	AI	-0.2	-0.7	580%	687%
	RA	-0.2	-0.9	304%	383%
	LP	-0.7	-4.0	311%	400%
Intra & Intra 1D & re-estimated motion	AI	-0.2	-0.7	580%	687%
	RA	-0.2	-0.6	485%	30548%
	LP	-0.8	-2.8	590%	39548%

Table 6.1 – *Coding performance of the SDec scheme exploiting different combinations of coding modes during the SDec process (Ref: HM12).*

scheme can be indeed further improved by adding more coding modes in the SDec competition.

Concerning the runtime, combining several coding modes in the SDec process increases unavoidably both the encoding and the decoding times. Particularly, the additional use of re-estimated motion increases significantly the runtime due to the computation of optical flow on the frame level.

Table 6.2 presents the selection rate for each coding mode used in the combination. The results are computed in LP configuration on the HEVC test set.

Combination of coding modes	% Mode 1	% Mode 2	% Mode 3
Intra & Intra 1D	5.0%	1.0%	
Intra & Intra 1D & Re-estimated motion	4.9%	0.9%	1.3%

Table 6.2 – *Selection rate for each coding modes in the SDec scheme that exploits a combination of coding modes during the SDec process, LP configuration.*

According to table 6.2, when combining all three coding modes in the SDec process, Intra is the coding mode most selected by the SDec competition to encode the current block (4.9%), followed by the re-estimated motion (1.3%) and finally the Intra 1D mode (0.9%).

Finally, we conduct the SDec scheme, which combines all three coding modes (Intra, Intra 1D and re-estimated motion) in the SDec process, on an extended test set with additional coding rates and test sequences. The test environment is summarized as follows:

Codec: HEVC, test model version 12.0 (HM12) QPs range: Low bit rate (LBR) QP = {27, 32, 37, 42}, Medium bit rate (MBR) QP = {22, 27, 32, 37}, Hight bit rate (HBR) QP = {17, 22, 27, 32} Test sequences: CTC test set for HEVC and additional sequences Evaluation: Average B-D rate on all frames

Sequences class	AI			RA			LP		
	LBR	**MBR**	**HBR**	**LBR**	**MBR**	**HBR**	**LBR**	**MBR**	**HBR**
Class A	-0.1	-0.1	-0.1	-0.1	-0.1	-0.2	-1.4	-0.9	-0.5
Class B	-0.2	-0.1	-0.1	-0.3	-0.2	-0.1	-1.1	-0.6	-0.3
Class C	-0.3	-0.2	-0.2	-0.2	-0.2	-0.1	-0.8	-0.5	-0.3
Class D	-0.1	-0.1	-0.1	-0.2	-0.1	-0.1	-0.7	-0.4	-0.2
Class E	-0.6	-0.5	-0.3	-0.5	-0.3	-0.2	-1.8	-1.4	-0.9
Class F	-0.5	-0.4	-0.4	-0.7	-0.5	-0.4	-1.5	-1.2	-0.9
Average	**-0.3**	**-0.2**	**-0.2**	**-0.3**	**-0.2**	**-0.2**	**-1.2**	**-0.8**	**-0.5**
Max gain	-1.2	-0.8	-0.7	-1.0	-0.8	-0.7	-3.6	-3.1	-2.6
BonneFranquette 2160p	-0.3	-0.2	-0.1	-0.3	-0.2	-0.2	-1.1	-0.8	-0.6
ParisFade 1088p	-0.1	-0.1	-0.1	-15.0	-13.1	-5.5	-9.6	-9.6	-6.1
BasketBall 1080p	-0.4	-0.7	-1.9	-0.2	-0.1	-0.2	-1.1	-0.6	-0.3
DespicableMeMoon 1080p	-0.2	-0.2	-0.2	0.1	0.0	-0.2	-2.2	-1.9	-1.4
RollingTomatoes 1080p	-2.3	-0.9	-0.3	-1.7	-1.5	-1.1	-2.7	-2.7	-1.2
S15Rugby 1080p	-0.1	-0.2	-1.5	0.0	0.0	-0.1	-0.7	-0.2	-0.1
Average	**-0.6**	**-0.4**	**-0.7**	**-2.8**	**-2.5**	**-1.2**	**-2.9**	**-2.6**	**-1.6**
Enc Time	580%			485%			590%		
Dec Time	687%			30548%			39548%		

Table 6.3 – *Bit rate savings in percentage of SDec scheme exploiting a combination of three coding modes (Intra, Intra 1D and re-estimated motion) during the SDec process (Ref: HM2)*

In table 6.3, we observe that for sequences containing complex texture ("BasketBall_1080p" and "S15Rugby_1080p"), the coding gain in AI configuration and HBR coding rate is significant (with gain of respectively -1.9% and -1.5%) thanks to Intra 1D mode being exploited. Unfortunately, if we compare the average gain obtained on the HEVC test set with the results in chapter 3 where only Intra mode is used, no increase in gain is made because Intra 1D is efficient only on very few particular test sequences as mentioned earlier. Nevertheless, combining several different coding modes in the SDec scheme has a major advantage of providing the flexibility to adapt to different screen content of test sequences.

6.2 SDec scheme exploiting several coding modes applied in 3D

In this section, the SDec scheme which exploits a combination of different coding modes is proposed for 3D coding. First, the description of the proposed scheme is detailed. Experimental results are then given.

6.2.1 Description

Compared to 2D coding, 3D coding differs in the fact that the base view must be coded first. Dependent views are then coded.

On the base view, the SDec scheme is exactly similar as in previous section related to 2D coding, taking the colocated block as the SDec reference.

On dependent views, we propose to apply the SDec scheme in two different ways corresponding to two methods for selecting the SDec reference. More precisely, supposing that we are encoding the current block in the frame at instant t_i in a dependent view as illustrated in figure 6.3, the SDec reference can be selected from following blocks:

– the temporal colocated block, denoted by *Col*, located in a reconstructed frame at instant t_j and in the same dependent view,
– or the block pointed by the disparity vector (DV) and located in the frame at instant t_i but in the base view, denoted by *IVref*.

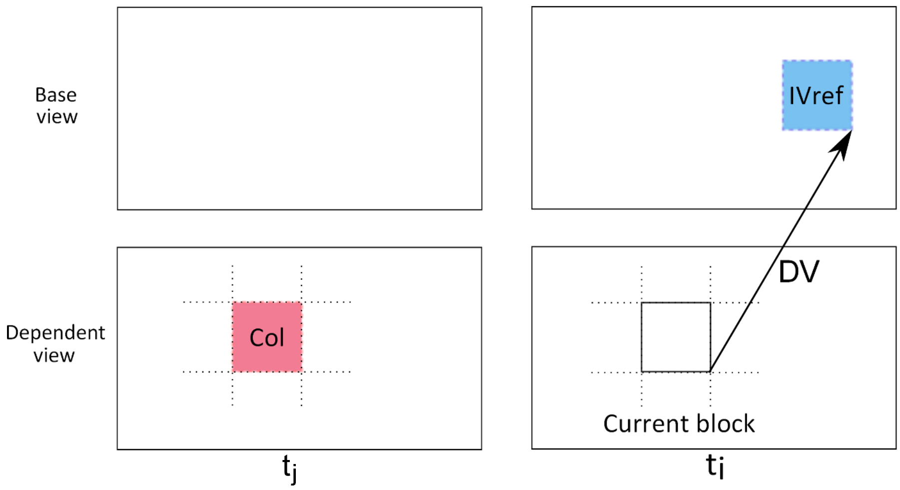


Figure 6.3 – *Two SDec reference candidates for the current block in a dependent view: the temporal Col in the same view or the inter-view IVref in the base view.*

In the experiments, we will evaluate separately Col and $IVref$ as the single candidate for the SDec reference in order to decide which block performs the best.

6.2.2 Experimental results

All experiments are conducted following common test conditions (CTCs) defined by JCT-3V (Rusanovsky et al.). Four standard QP combinations for texture and depth are considered: (25;34), (30;39), (35;42) and (40;45). The test set contains eight sequences defined in CTCs.

Test environment can be summarized as follows:

Codec: 3D-HEVC, test model version 9.3 (HTM9.3)
QPs range: Medium bit rate with four QP combinations for texture and depth: (25;34), (30;39), (35;42) and (40;45).
Test sequences: CTC test set for 3D-HEVC, shortened to 1 second
Evaluation: Average B-D rate on all frames

First, we propose to evaluate the performance of each of two proposed candidates to be the SDec reference: Col and $IVref$. Various tests exploiting separately different coding modes (Intra, Intra 1D, re-estimated motion) are conducted. The experimental results are given in table 6.4. The columns $V0$, $V1$ and $V2$ show the texture coding gain respectively on the base view and two side dependent views. The "Video total" column gives results on all coded views. Result in the "Synt." column provides gain on all synthesized views. Note that the column $V0$ related to the base view is common for both *Col* and $IVref$ since those two candidates only affect dependent views.

		Col				IVref			
	V0	**V1**	**V2**	**Video total**	**Synt.**	**V1**	**V2**	**Video total**	**Synt.**
Intra mode	-0.4	0.3	0.1	-0.2	-0.2	-0.1	-0.1	-0.3	-0.2
Intra 1D mode	-0.2	0.2	-0.3	-0.2	-0.2	0.0	-0.2	-0.2	-0.1
Re-estimated motion	-0.1	0.2	-0.2	0.0	0.2	-0.3	-0.8	-0.2	0.0

Table 6.4 – *Comparison between temporal Col and inter-view IVref blocks as candidates to be the SDec reference (Ref: HTM9.3).*

We observe that on dependent views, $IVref$ performs better than *Col* as block candidate for the SDec reference in most of the cases. This can be explained as $IVref$ is often more similar to the current block than *Col* because of better inter-views correlation compared to temporal correlation.

In the next step, using $IVref$ as the single candidate for the SDec reference on dependent views, we then conduct different experiments corresponding to different combinations of coding modes exploited during the SDec process. The experimental results are given in the table 6.5.

	V0	**V1**	**V2**	**Max.**	**Video total**	**Synt.**	**EncTime**	**DecTime**
Intra	-0.4	-0.1	-0.1	-1.0	-0.3	-0.2	147.3%	134.0%
Intra 1D	-0.2	0.0	-0.2	-1.5	-0.2	-0.1	344.0%	166.8%
Re-estimated motion	-0.1	-0.3	-0.8	-2.9	-0.2	0.0	302.3%	37097.3%
Intra & Intra 1D	-0.3	-0.3	-0.4	-1.8	-0.4	-0.3	381.7%	215.6%
Intra & Intra 1D & Re-estimated motion	-0.4	-0.3	-0.4	-1.5	-0.4	-0.3	579.1%	23947.9%

Table 6.5 – *Coding performance of the 3D SDec scheme using different combinations of coding modes during the SDec process (Ref: HTM9.3).*

We observe that combining different coding modes during SDec process increases indeed the coding performance compared to the separate use of each coding mode. The combination of Intra and Intra 1D modes yields -0.3%, -0.3% and -0.4% respectively on V0, V1 and V2. The maximum gain achieved (among V0, V1 or V2) is -1.8%. Average gains in coded views and synthesized views are respectively -0.4% and -0.3%.

Combining all three modes (Intra, Intra 1D and re-estimated motion) increases slightly the coding performance, yielding -0.4%, -0.3% and -0.4% respectively on V0, V1 and V2. However, additionally using re-estimated motion requires the computation of optical flow between reconstructed frames, increasing thus both encoding and decoding time.

We also give the selection rate for each coding modes in case the combination of Intra, Intra 1D and re-estimated motion is used during the SDec process. According to table 6.6, in the base view, Intra mode is most selected (4.9%), followed by Intra 1D (0.6%) and re-estimated motion (0.2%). In dependent views, Intra is again the coding mode selected the most (0.9%).

We conclude finally that increasing the number of coding modes in the SDec scheme improves indeed the coding performance. The increase in gain is however limited considering the considerably longer runtime

	Base view	**Dependent views**
% Intra	4.9%	0.9%
% Intra 1D	0.6%	0.3%
% Re-estimated motion	0.2%	0.3%
% Total SDec	5.7%	1.5%

Table 6.6 – *Selection rate for each coding modes being used in the SDec scheme combining several coding modes during the SDec process.*

since more coding modes are tested on the SDec reference during the SDec process.

6.3 Improvement using adaptive selection depending on the visual content

The limited results obtained can be explained by following reasons:

- The choice of the SDec reference is still simple since it is based on the texture similarity between the current block and the SDec reference.
- The coding modes exploited in the SDec process are not suited for the visual content of the encoded sequence.

In this section, we will demonstrate that selecting the SDec reference based only on texture similarity is not always the good choice and furthermore, the coding modes used in the SDec process should be selected depending on the visual content of the sequence.

For simplification purpose, we consider only the Intra coding mode to be used during the SDec process. Since the SDec mechanism is to inherit the optimal mode of the SDec reference and to apply it for the current block, it is required that the SDec reference could give the same optimal mode as for the current block. There are thus two preferable ways of selecting block to be the SDec reference:

- The SDec reference has identical texture as the current block in terms of block distortion.
- The SDec reference has only the same texture direction as the current block but not exactly identical in terms of block distortion.

We conduct a test in order to evaluate which selection method is more favorable. Let two following blocks be the SDec reference:

- Colocated block.
- If temporal prediction modes (Skip, Merge, Inter) perform better than Intra for the current block, the MV is exploited to point to a block that will be used as the SDec reference. In other term, this MV points to a block that minimizes the distortion with current block and is denoted by $MV_{minDist}$.

We propose to apply the SDec scheme to encode the test sequence "$paris\_fade\_1920 \times 1088$" containing a particular fading effect as depicted in figure 6.4.

Table 6.7 presents the SDec performance in two cases of selecting block candidate for the SDec reference. Compared with the use of the colocated block, if the SDec reference is the block pointed by $MV_{minDist}$, its distortion with the current block (computed using a HEVC metric that is based on

Figure 6.4 – *Test sequence "Paris_fade" with fading effect: the illumination is gradually decreased as seen from left to right.*

the difference regarding luminance and chrominance components) is indeed lower. In other terms, there are more similarity regarding the block distortion when using $MV_{minDist}$ than when using the colocated block. However, the gain of the SDec scheme is -1.1% with 36.1% of total blocks encoded in SDec mode, much more significant than when using $MV_{minDist}$ which yields only -0.1% with 2.7% of total blocks encoded in SDec mode. This is because the colocated block can still provide the same Intra direction despite the change of illumination, given that the block texture remains the same.

SDec reference	**Block pointed by $MV_{minDist}$**	**Colocated block**
Avg. distortion (using HEVC metric)	2044×10^6	2102×10^6
Gain	-0.1%	-1.1%
%SDec	2.7%	36.1 %

Table 6.7 – *SDec performance on test sequence* $paris\_fade\_1920 \times 1088$ *with different block candidate as the SDec reference (Ref: HM12).*

Another test is conducted to demonstrate that coding modes in the SDec scheme should be exploited depending on the sequence content. We propose to encode the sequence "BasketBall_1080p". As illustrated in figure 6.5, the visual content of the sequence is divided by different regions that contain particular characteristics and are highlighted in different colors as follows:

– Red region contains complex motion (with basketball players moving around),
– Green region contains complex texture in translational movement (the playing surface of the basketball court),
– Remaining region is not highlighted.

For the SDec scheme, we use only a single candidate, computed using the template matching technique (cf. 3.2.2), for the SDec reference. Coding modes that are put in the SDec competition are pre-defined for CUs in different regions. For the region that is not highlighted, all three coding modes Intra, Intra 1D and Inter with re-estimated motion are exploited. For the region with complex motion, only Intra mode is used. For the region containing complex texture in translational movement, Intra 1D and Inter with re-estimated motion are exploited.

A coding gain of -0.1% is observed while other experiments, in which a same configuration of coding modes (separately or in combination) is exploited in the SDec scheme without distinguishing different particular

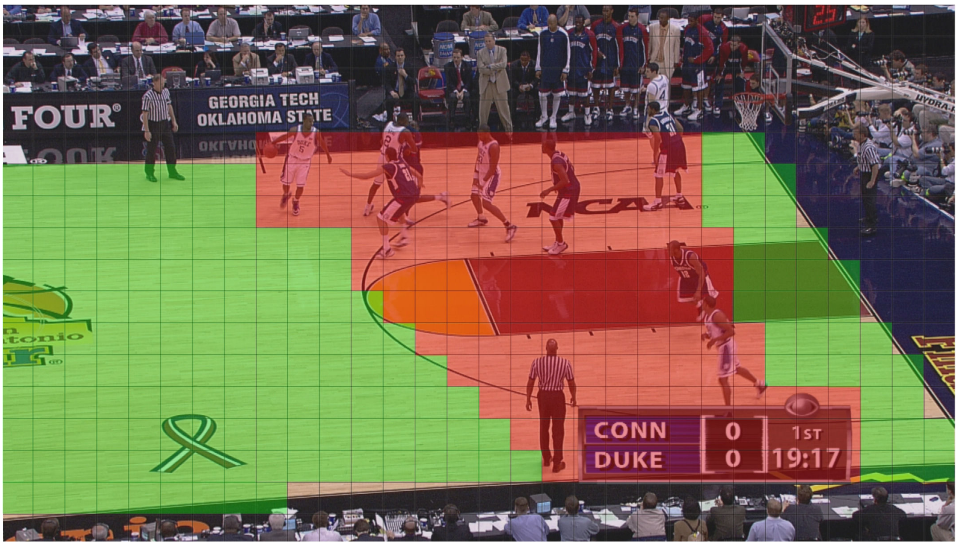


Figure 6.5 – *Test sequences "BasketBall_1080p" with visual content divided in particular regions highlighted in different colors: red region contains complex motion, green region contains complex texture in translational movement.*

regions in the sequence, do not provide gain. An adaptive selection of coding modes depending on the visual content of the encoded sequence is therefore considered as an interesting improvement for the SDec scheme.

In conclusion, given the complex problematic of selecting block candidates for the SDec reference, which should not be selected simply based on the distortion minimization, and of exploiting suitable coding modes in the SDec process for different characteristics of sequence content, it is proposed to use more sophisticated techniques, such as machine learning, to provide an improvement based on adaptive selection: by exploiting data from previously decoded frames, a learning technique can automatically derive the visual characteristic of the current block, allowing thus to select efficiently the SDec reference and the coding modes used in the SDec process. For example, for region containing complex texture in a translational movement, Intra 1D can be used on a block selected as the SDec reference using the template matching technique. For visioconference sequences containing little movement, Inter with re-estimated motion can be exploited on the colocated block which is taken as the SDec reference.

Conclusion

In this chapter, it is proposed that a multitude of coding modes are used during the SDec process. The competition on the SDec reference includes Intra mode, Intra 1D mode and re-estimated motion. Experimental results of the practical application in 2D shows that adding different coding modes in the SDec competition helps to better encode more sequences having particular characteristics, despite that no improvement in average gain for the standard test set is observed.

In 3D coding, using an inter-view block as the SDec reference, it is observed that combining more coding modes in the SDec scheme gives better coding performance, resulting in improved gain in base view and dependent views compared to the separate use of each coding mode.

Lastly, an improvement for the SDec scheme is presented, proposing to exploit learning techniques to provide an adaptive selection of the SDec reference and the coding modes being used during the SDec process depending on the visual content of encoded sequences. This perspective motivates us to conduct research in order to apply machine learning in video coding, a concept that is described in more detail in chapter 8.

Part 2

Machine learning based video coding

7 Improvement for Intra MPM signaling scheme

In this chapter, a modified signaling scheme for the Intra Most Probable Mode (MPM) is proposed in order to signal more efficiently the Intra directions for the classic Intra mode. First, the basic idea and preliminary observations are given. The description of the proposed method and the experimental results are then presented. Finally, a perspective for improvement is detailed, introducing the idea to exploit the machine learning concept that makes predictions based on observed data.

7.1 Basic idea

In HEVC test model software, the Intra direction of a block encoded in Intra mode is signaled in the bit stream. The Intra MPM signaling scheme is used to reduce the signaling cost of the Intra direction. The scheme consists in predicting three most probable directions which are signaled using only one or two bits if one of the predicted values matches the Intra direction of the block. We propose a modified MPM signaling scheme which allows to replace one among three MPM values or to add a fourth MPM value. For the replacement, no modification is made concerning the signaling syntax elements. For the adding of the fourth MPM value, one bit is simply inserted to signal the additional value.

7.2 Preliminary observations

The preliminary tests are conducted to obtain some statistics on the Intra MPM values. We display information concerning three values of MPM and the optimal Intra direction of the current PU. In order to evaluate the efficiency of every Intra direction candidate for the current PU without any bias caused by the MPM scheme, each of all 35 directions is signaled equally using 6 bits and without the MPM scheme. Moreover, the cost of signaling the Intra direction is also not accounted.

The optimal Intra direction and three MPM values of every block are stored like in table 7.1. We also display, for each PU, the corresponding case among 5 Intra MPM cases that are defined depending on the Intra directions of the Above (A) and Left (L) PUs:

- Case 1: $A = L$, both are angular
- Case 2: $A = L$, both are not angular
- Case 3: $A \neq L$, both are not Planar
- Case 4: $A \neq L$, with A or L is Planar and other not DC
- Case 5: A, L are Planar and DC

PU	Optimal Intra dir.	MPM1	MPM2	MPM3	MPM case
0	0	0	1	26	2
1	1	0	1	26	2
2	1	1	0	26	2
3	0	0	1	26	5
...	...	...	...	...	...

Table 7.1 – *Example of a table displaying data related to Intra MPM values for encoded blocks.*

In table 7.1, we can observe, for example, that the first PU has Planar (index 0) as the optimal Intra direction, while the third PU has DC (index 1). The three values of MPM for the first PU are respectively Planar, DC and Vertical (index 26). The corresponding MPM case is 2.

Let us consider following candidates susceptible to be replaced or added in the MPM signaling scheme:

Planar, DC, HOR, VER, MPM1-1, MPM1+1, MPM2-1, MPM2+1, MPM3-1, MPM3+1 where MPMi-1 and MPMi+1 are respectively the Intra directions on the left and on the right of MPMi among all angular Intra directions.

Based on the table 7.1, we can observe, for each of the five MPM cases, whether or not the optimal Intra direction is correctly predicted by a MPM value or a candidate previously proposed. The table 7.2 summarizes how many times each HEVC MPM candidate corresponds to the optimal Intra mode to be encoded (with values a_1, a_2, a_3 respectively for MPM1, MPM2 and MPM3), and how many times each proposed MPM candidate would correspond to the optimal Intra mode to be encoded (with value x).

	HEVC MPM cand.			**Proposed MPM cand. (x)**									
	MPM1 (a1)	**MPM2 (a2)**	**MPM3 (a3)**	**Planar**	**DC**	**VER**	**HOR**	**MPM1-1**	**MPM1+1**	**MPM2-1**	**MPM2+1**	**MPM3-1**	**MPM3+1**
C1	6858	4494	4441	2344	1902	1165	1161	N/A	N/A	2636	120	120	2541
C2	9276	1679	717	N/A	N/A	N/A	620	N/A	N/A	N/A	N/A	401	353
C3	49062	48303	58612	N/A	33387	27064	25818	31939	32252	30846	29422	N/A	N/A
C4	16431	14137	9721	N/A	N/A	4795	4582	3049	3030	2932	2875	N/A	N/A
C5	3787	2437	769	N/A	N/A	N/A	632	N/A	N/A	N/A	N/A	416	346

Table 7.2 – *Statistics on the number of times that conventional and proposed MPM candidates match the optimal Intra direction of the current block.*

7.3 Proposed method and experimental results

In this section, following test environment is used for all the experiments:

Codec: HEVC, test model version 10.1 (HM10.1)
QPs range: Medium bit rate QP = {22, 27, 32, 37}
Test sequences: CTC test set for HEVC
Evaluation: Average B-D rate on all frames in AI

Based on the table 7.2 made during preliminary tests, we observe that the replacement can only be successful when x is greater than a_1, a_2 or a_3. According to the table, there is no candidate among the proposed ones that performs better than three conventional MPM candidates. Thus, it is not interesting to replace MPM values by proposed candidates.

Tests where a MPM value is replaced by proposed candidates are conducted as confirmation purpose. As expected, no gain has been obtained.

For the addition of a fourth MPM value, a quick study shows that:
– when adding in 3^{rd} or 4^{th} position, the scheme can be successful if $2x > a_3$
– when adding in 2^{nd} position, the scheme can be successful if $3x > a_2 + a_3$
– when adding in 1^{st} position, the scheme can be successful if $4x > a_1 + a_2 + a_3$

According to the table 7.2, we observe that there exist candidates that are susceptible to be added in the fourth position. For each MPM case, the following candidates are considered (in this order, in case of equality with another MPM candidate):
– Case 1: MPM2-1, MPM3+1, Planar
– Case 2: HOR, MPM3-1

– Case 3: DC, MPM1+1, MPM1-1, MPM2-1, MPM2+1
– Case 4: VER, HOR
– Case 5: HOR, MPM3-1

The modified MPM signaling scheme that take account of the fourth MPM candidate is described in figure 7.1:

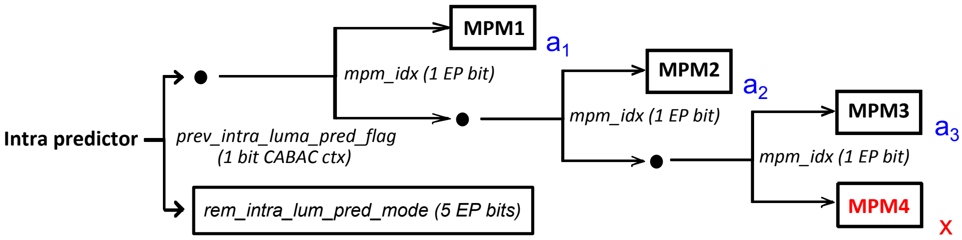


Figure 7.1 – *Signaling scheme of the proposed method adding a fourth MPM value.*

The average gain of this approach for AI configuration is -0.1%, with a maximum gain of -0.2%. The increase in runtime is insignificant.

The following table 7.3 shows the percentage of each MPM case (column 2), how many times one of the MPM value fits with the optimal Intra mode to be encoded (column 3), and the details for each MPM (columns 4, 5, 6, 7). The statistics of the proposed method are reported in brackets to be compared with the statistics of the reference HM. We observe that the MPM efficiency for all MPM cases is indeed increased thanks to the proposed method.

Case	% appearance	Efficiency of MPM scheme	Efficiency of each MPM candidate			
			MPM1	MPM2	MPM3	Added MPM4
C1	11.6% (12.6%)	72.2% (77.0%)	62.8% (61.7%)	18.7% (17.3%)	18.5% (11.9%)	0% (9.0%)
C2	9.6% (9.5%)	71.6% (76.8%)	58.3% (54.6%)	26.7% (25.4%)	15.1% (9.9%)	0% (10.1%)
C3	49.3% (52.0%)	75.4% (82.6%)	50.9% (48.2%)	27.9% (26.9%)	21.2% (13.6%)	0% (11.4%)
C4	20.7% (17.8%)	66.5% (72.2%)	51.0% (48.6%)	29.7% (29.4%)	19.3% (11.6%)	0% (10.4%)
C5	8.8% (8.1%)	73.3% (78.1%)	54.5% (51.4%)	31.1% (30.0%)	14.4% (9.5%)	0% (9.1%)
All	100%					

Table 7.3 – *Statistics on the efficiency of proposed MPM scheme (in brackets) compared to the reference HM.*

Considering the above results, we propose to adapt, on a frame basis, the five proposed lists of added candidates corresponding to five MPM cases, in order to benefit from more flexibility. For each MPM case, one new candidate is computed based on the previous frame: it is counted the number of times where, for each decoded Intra mode, none of the four MPM has not been used. Most likely, it would correspond to an efficient MPM candidate.

For this configuration, experimental results show the same average gain of -0.1%, with a maximum gain of -0.6%. The gain is mainly for class F sequences (-0.4%).

Finally, a simpler model is tested, proposing to consider the fourth MPM candidate selected from a predefined list. We first follow the HEVC process to establish the conventional MPM list. A fourth MPM candidate is then added, with the following priority order, whatever the case:

Planar, DC, VER, HOR, MPM1-1, MPM1+1, MPM2-1, MPM2+1, MPM3-1, MPM3+1

Experimental results are given in table 7.4. Both the configurations AI 8 bits and AI 10 bits are tested. The proposed method is tested not only on the standard HEVC test set, but also on several additional test sequences with different resolutions.

	AI 8 bits			**AI 10 bits**		
	Y	U	V	Y	U	V
Class A	-0.1	-0.1	-0.1	-0.1	-0.1	-0.1
Class B	-0.1	-0.1	-0.1	-0.1	-0.1	-0.1
Class C	-0.1	-0.1	-0.2	-0.1	-0.2	-0.1
Class D	0.0	-0.1	-0.1	0.0	-0.1	-0.1
Class E	-0.2	-0.2	-0.2	-0.2	-0.1	-0.1
Class F	-0.2	-0.2	-0.2	-0.2	-0.3	-0.2
Overall	**-0.1**	**-0.1**	**-0.1**	**-0.1**	**-0.1**	**-0.1**
Max.	-0.3			-0.3		
Min.	0.0			0.0		
Extra Sequences	-0.2	-0.1	-0.1	-0.2	-0.1	-0.1
Max.	-0.8			-0.6		
Min.	0.0			0.0		
EncTime	101%			101%		
DecTime	101%			101%		

Table 7.4 – *Coding results of proposed method adding a fourth MPM value, AI configurations 8 bits and 10 bits (Ref: HM10.1).*

Other configurations (LP, RA) are also tested. The results are given in table 7.5:

Both tables 7.4 and 7.5 show minor yet consistent gains, especially for AI configuration. An average gain of -0.1% is obtained in AI and LP configurations, while no loss is observed in RA configuration. Given that this modification comes at no cost, it is considered as an interesting improvement..

7.4 Perspectives

Instead of using predefined values as candidates to add in the MPM signaling scheme, an interesting perspective is to adaptively calculate a most probable direction based on the previously decoded frames. For example, it might be possible to predict the optimal Intra direction of the current block based on the spatial distribution of Intra blocks and their optimal direction in previously decoded frames. We propose as follows a test that gives some preliminary observations.

	LP			RA		
	Y	U	V	Y	U	V
Class A	0.0	0.0	0.0	0.0	0.1	-0.1
Class B	0.0	0.0	0.1	0.0	0.0	0.0
Class C	0.0	-0.2	0.2	0.0	-0.1	-0.2
Class D	0.0	-0.1	0.2	0.0	0.1	-0.1
Class E	-0.2	0.5	-1.1	0.0	-0.2	-0.3
Class F	-0.1	0.0	-0.3	-0.1	-0.1	-0.1
Overall	**-0.1**	**0.0**	**-0.1**	**0.0**	**0.0**	**-0.1**
Max.	-0.1			-0.1		
Min.	0.2			0.1		
EncTime	101%			98%		
DecTime	101%			98%		

Table 7.5 – *Coding results of proposed method adding a fourth MPM value, LP and RA configurations (Ref: HM10.1).*

In order to visualize the correlation between the optimal Intra direction of the current block to encode and the spatial distribution of Intra blocks found in already reconstructed frames, we propose to display the optimal Intra direction of:

– Intra blocks in the current frame,
– Intra blocks (overlapped included) in the previously decoded frame.

The optimal Intra direction is computed using the exhaustive R-D competition on corresponding blocks. To ensure that there is no bias from the Intra MPM signaling scheme, the competition does not include the signaling cost of the Intra direction which is not transmitted in the bit stream.

In the figure 7.2, we visualize all blocks having Planar (marked by a red dot in the center) and DC (marked by a green dot in the center) as the optimal Intra direction. Note that in the previously decoded frame at instant t_{i-1}, overlapped blocks are also considered, leading to a higher density of colored dots than in the current frame at instant t_i. We observe that there are some correlations in the positions of Planar and DC blocks between frames at instants t_{i-1} and t_i. For example, many Planar blocks are gathered within the region of the horse rider.

t_{i-1} t_i

Figure 7.2 – *Blocks having Planar (red) or DC (green) as optimal Intra direction in two consecutive frames, sequence RaceHorses_832 × 480, QP 37.*

In figure 7.3, all blocks having the indice of Intra direction among values of 7, 8, 9, 10 and 11 as the optimal Intra direction are highlighted.

Different colors are used: blocks having optimal Intra direction with indice respectively of 7, 8, 9, 10 or 11 are marked in the center by red, green, blue, cyan or olive dots. Similarly as in figure 7.2, we observe that the distribution of Intra blocks depending on their optimal Intra direction is correlated between frames at instants t_{i-1} and t_i.

Figure 7.3 – *Blocks having directions 7 (red), 8 (green), 9 (blue), 10 (cyan) or 11 (olive) as optimal Intra direction in two consecutive frames, sequence* $ChinaSpeed\_1024 \times 768$*, QP 22.*

Those preliminary observations introduce the idea to exploit data collected from the previously decoded frames to make predictions on the current frame. Machine learning is a technique which fits for that purpose. Our remaining works are dedicated to make use of machine learning concept in video coding, and will be presented in the next chapter.

Conclusion

We have studied in this chapter a method which aims to improve the Intra MPM signaling scheme. Several MPM modifications were tested, including the replacement of one MPM candidate or the adding of a fourth MPM candidate. Eventually, adding an additional MPM candidate gives slight but systematic gains (0.1%), especially in AI configuration. Negligible increase in runtime is observed.

In perspective, it is proposed to adaptively compute a most probable MPM candidate instead of using predefined values, introducing the idea to exploit machine learning techniques to make predictions based on data from previously decoded frames.

8 Video coding scheme based on machine learning

In this chapter, we propose a novel video coding scheme that exploits machine learning techniques to improve the compression ratio by reducing the signaling overhead. First, we present the background of our works, by describing the generalities of conventional coding schemes and machine learning based coding schemes that are proposed in the state of the art. A general explanation on different types of histograms used by machine learning techniques is also given. Then, the proposed coding method is presented. Next, practical applications with experimental results are detailed to illustrate the proposed method. Finally, we discuss about some possible perspectives to improve the coding performance.

8.1 Background

8.1.1 Video coding methods based on machine learning

In conventional codecs (MPEGx, H.264, HEVC,...) the encoding mode used to encode a block is signaled in the bit stream. In general, this mode is computed at the encoder side by using the rate-distortion (R-D) criterion. Unlike conventional codecs, there are many coding methods using machine learning, already presented in the chapter 1 (Han et al. 2010, Kalva and Christodoulou 2007, Jillani and Kalva 2009, Carrillo et al. 2010, Ma et al. 2009, Lampert 2006, Tohidypour et al. 2014, Zhou et al. 2009, Di and Yuan 2010, Chiang et al. 2011, Xiong and Li 2012), that exploit learning techniques to predict the optimal mode of a block based on its characteristics (also called *features*). Different features of the block (average grayscale value, size, grayscale histogram, etc.) can be considered as features. Those approaches can be assimilated to a classic classification problem which consists of two major processes: a *training step* and a *classification step*, as depicted in figure 8.1.

a) Training step: compute a block classifier based on the training data.
A training set is created and includes all blocks (i.e. *training samples*) located in already decoded frames. Each block is characterized by its feature and its R-D optimal coding mode. A classification algorithm (e.g. decision tree, SVM, K-nearest neighbors) is conducted on the set of couples *(Feature, Mode)* of those blocks to infer a block classifier. In the end, the optimal classifier that best classifies training blocks by their coding mode based on their feature is obtained.

b) Classification step: predict the optimal R-D coding choice of the current block to encode using the block classifier.
For the current block being encoded, its feature is first computed. The block classifier obtained in the training step is then used to predict the R-D optimal coding choice for the current block based on its feature.

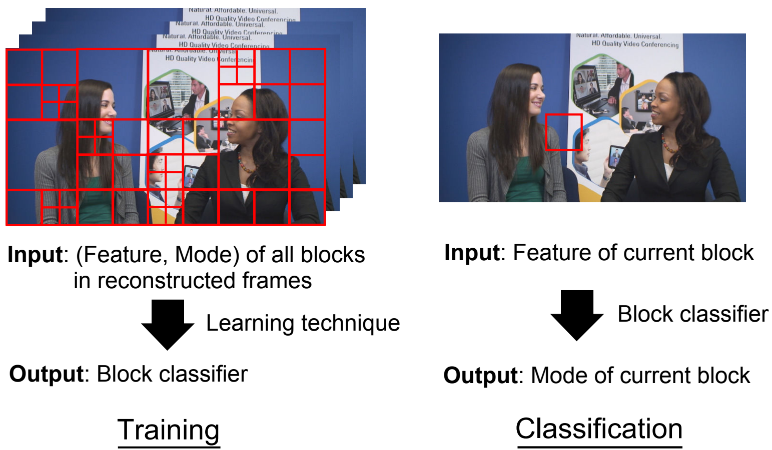


Figure 8.1 – *Classification problem applied to video coding, with the training step on the reconstructed frames (e.g. frames at instants t_{i-1}, t_{i-2}, etc.) and the classification step on the current frame at instant t_i.*

All the coding methods based on machine learning proposed in the literature are designed for complexity reduction. Indeed, the coding mode of a block is selected directly using classification techniques rather than performing the exhaustive R-D competition. Those methods do not reduce the signaling overhead. They only help the encoder to make a fast choice based on observations from the past. That choice must still be signaled in the bit stream.

None of these methods uses the training step or the classification step at the decoder side: the decoder for these methods reads the instruction in the bit stream and then applies the corresponding mode to decode a block, remaining thus a standard decoder, such as for HEVC.

Figures 8.2 and 8.3 illustrate respectively the encoders and the decoders corresponding to conventional codecs and codecs exploiting machine learning. The following notations are used:

- P: the current block to encode/decode
- ϵ: the residual texture
- M_X^*: the optimal coding mode of the block X
- X_X: the characteristics of the block X, i.e. parameters that characterize X. For example: the average gray level, the block size, the magnitude of the motion vector, the direction of the motion vector, etc.
- M_X^{**}: the probable coding mode of the block X
- f: the block classifier, modeled as a function that takes the characteristics of a block as arguments and gives as output the most probable coding mode. This classifier is predetermined a priori through a training step using machine learning techniques (e.g. support vector machines, decision tree, etc.) conducted on a set of training block samples independent to the current sequence to encode

For the conventional HEVC encoder depicted in figure 8.2 (a), a competition of all coding modes is conducted on the current block P to find its optimal mode M_P^*. The texture residual ϵ along with M_P^* are signaled in the bit stream.

At the decoder depicted in figure 8.3 (a), the optimal mode M_P^* and the residual texture ϵ must be read from the bit stream in order to decode the block P.

For the encoder based on machine learning depicted in figure 8.2 (b), a classification step is carried out in order to predict the most probable mode of P. This procedure consists of two phases: the calculation of features X_P of P, then the computation of the probable mode M_P^{**} by applying the block classifier f on X_P. This most probable mode M_P^{**} is used to encode P, and then is signaled in the bit stream. By comparing figure 8.2 (a) and (b), it is observed that the exhaustive R-D competition of coding modes on P is replaced by the classification step.

At the decoder depicted in figure 8.3 (b), the most probable mode M_P^{**} and the texture residual ϵ are read from the bit stream in order to decode P. Both the decoders illustrated in figure 8.3 (a) and (b) are therefore identical.

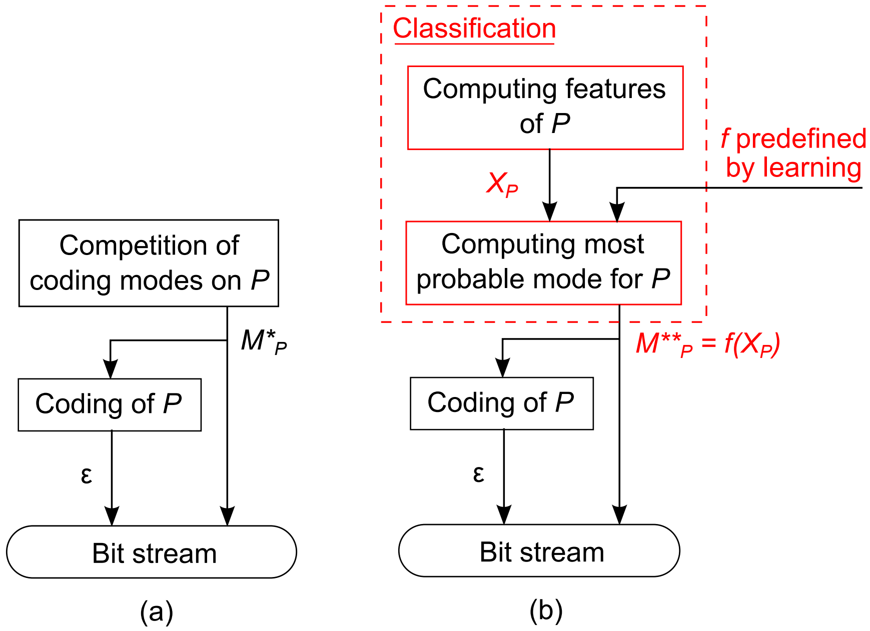


Figure 8.2 – *Conventional encoder scheme (a) and encoder scheme based on machine learning from literature (b).*

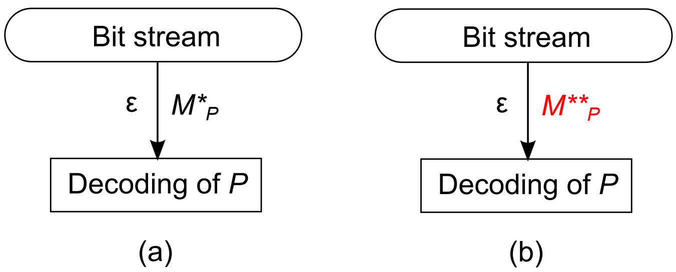


Figure 8.3 – *Conventional decoder scheme (a) and decoder scheme based on machine learning from literature (b).*

8.1.2 Exploitation of histograms as block features

A histogram is a function m_i that counts the number of observations that fall into each of the disjoint categories (known as bins). Thus, if we let n be the total number of observations and k be the total number of bins, the histogram m_i meets the following conditions:

$$n = \sum_{i=1}^{k} m_i.$$

Histograms are fast to compute, space efficient, and robust to noise. Therefore, they are a widely used tool in computer vision and pattern recognition community to represent, analyze and characterize various visual inputs. There are various methods to exploit histograms. We describe some common methods in the following sections.

8.1.2.1 Using histogram as a local descriptor of blocks

There are various types of histograms that can be exploited in order to locally describe blocks in a picture. The most basic one is the grayscale histogram which represents the distribution of the pixels in the image over the graylevel scale.

For the histogram of oriented gradients (HOG) (Dalal and Triggs 2005), it captures the shape of structures in the block region by computing the gradients over different directions. HOG descriptors are based on gradient angle and magnitude distributions. Considering an image I and gradient estimation filters $h_x = [-1, 0, 1]$ and $h_y = [-1, 0, 1]^T$, let g_x and g_y represent the gradient images generated by $g_x = I * h_x$ and $g_y = I * h_y$ where * represents the convolution operation. The magnitude of the gradient at each pixel can be calculated as:

$$G(i, j) = \sqrt{g_x(i, j)^2 + g_y(i, j)^2}$$

and the dominant gradient angle at each pixel can be estimated by:

$$\theta(i, j) = tan^{-1} \frac{g_y(i,j)}{g_x(i,j)}$$

HOG features can then be generated entirely on the basis of the gradient magnitude G and gradient angles θ at each pixel.

Texture variation in picture can also be captured by using the histogram based on Local Binary Patterns (LBP) (Ojala et al. 1994; 1996). LBP labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number, as illustrated in figure 8.4. Formally, given a pixel at position (x_c, y_c), the resulting LBP can be expressed in the decimal form as:

$$LBP(x_c, y_c) = \sum_{n=0}^{7} s(i_n - i_c) 2^n$$

where n runs over the 8 neighbors of the central pixel, i_c and i_n are the gray level values of the central pixel and the surrounding pixel, and $s(x)$ equals to 1 if $x = 0$ and equals to 0 otherwise. This type of histogram is known for its robustness to monotonic grayscale changes caused, for example, by illumination variations.

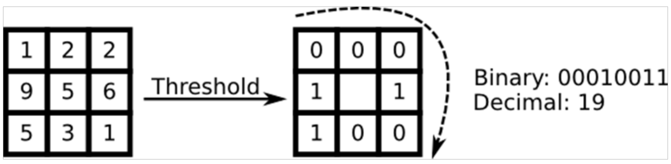


Figure 8.4 – *Grayscale constrast in LBP.*

Another interesting method is to exploit the histogram based on SIFT (Lowe 1999), which is an algorithm in computer vision to detect and describe local features in images. Blobs or corners in a picture are detected to be used as points of interest, which in turn are considered as features.

8.1.2.2 Using Bag-of-Feature framework as a global descriptor

Bag-of-Feature (BoF) is one of the popular visual features descriptors used for visual data classification. BoF is inspired by the concept Bag-of-Words that is used in document classification. A bag of words is a sparse vector of occurrence counts of words; that is, a sparse histogram over the vocabulary. In computer vision, a bag of visual words of features is a sparse vector of occurrence counts of a vocabulary of local image features.

BoF typically consists of two main steps. The first step is to obtain the set of bags of features. This step is actually an offline process. We can obtain set of bags for particular features and then use them for creating BoF descriptor. The second step is to cluster the set of given features into the set of bags created in the first step and then create the histogram taking the bags as the bins. This histogram can be used to classify an image or video frame.

Figure 8.5 illustrates a BoF descriptor using SIFT feature, with the two steps described as above. This is an example of classifying different types of shoes. In the first step, distinct visual characteristics of a large set of shoes are learned to form "bags of features" about shoes: from a large set of images of shoes, the SIFT feature points of all the images are extracted and the SIFT descriptor for each feature point is calculated; then, the set of feature descriptors is clustered to finally obtain the visual vocabulary about shoes. In the second step, a given type of shoes is classified using its BoF descriptor: the SIFT feature points of the given image of shoes are extracted and the SIFT descriptor for each feature point is calculated; then, by matching the descriptors with the vocabulary created in the first step, a histogram is built and is used for classifying the type of the given shoes.

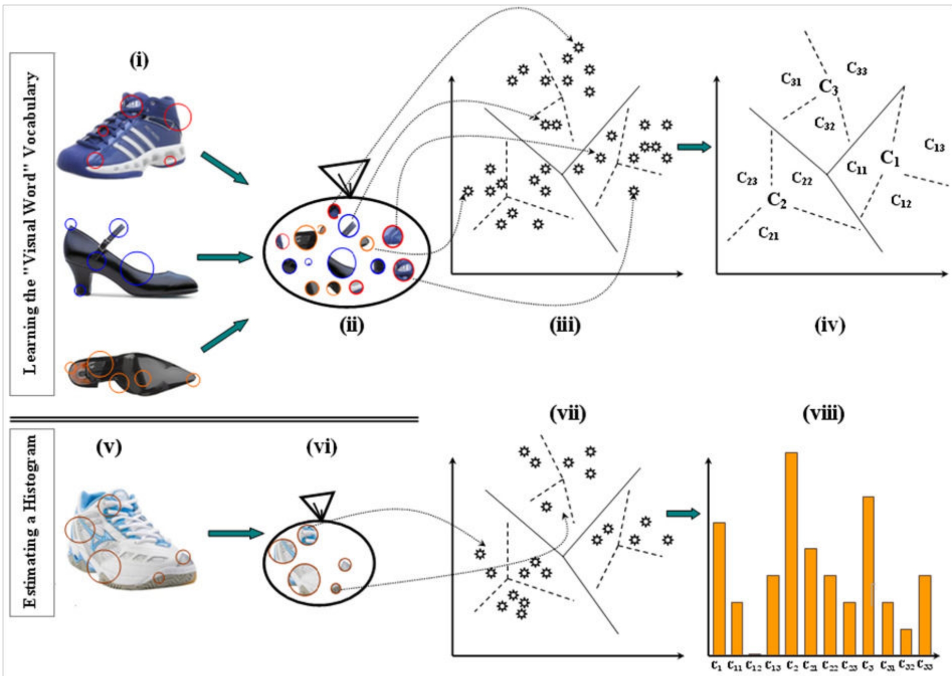


Figure 8.5 – *BoF descriptor with SIFT feature (Source: (Tomasik et al. 2009)).*

8.2 Proposed method

8.2.1 General description

Being based on classic classification problem, the proposed method includes both the training and classification steps as mentioned in previous section. The major difference compared to the existing machine learning based coding approaches is that proposed method does not have to signal the mode predicted by the classification process since that mode can be computed at the decoder side identically as at the encoder side, reducing thus the signaling overhead.

We present the proposed video coding scheme based on machine learning, by describing the encoding and decoding processes. Following notation is added:

– X'_P: causal characteristics of the block P, calculated on already decoded regions of picture (e.g. on already reconstructed neighboring blocks of P). They can be for example the average gray level, the block size, etc. of the surrounding blocks.

The encoder corresponding to the proposed method, illustrated in figure 8.6, includes the following steps:

Step 1: Classifying P

– Calculation of causal characteristics X'_P of P
– Deduction of probable coding modes of P, by applying the classifier f, predefined a priori from decoded blocks as training samples, on X'_P. The result $f(X'_P)$ of this classification includes in general the list of several probable coding modes of P

Step 2: Processing and extraction of a coding mode M_P^{**} from $f(X'_P)$ that includes the list of probable coding modes of P and their probability.

Step 3: Encoding P with the mode M_P^{**}. The signaling of M_P^{**} in the bit stream is optional and depends on the processing and extraction module as mentioned below. The texture residual ϵ resulting from the difference between original and reconstructed blocks is also signaled.

Given that the classification provides in general the list of several probable coding modes of P, the processing and extraction module allows:

(a) Either to modify the list of probable modes by reordering them in function of their probability (based on the result of the classification), and to extract modes that are most probable,

(b) Either to modify the list of probable modes by removing modes with least probability (based on the result of the classification), and to extract modes that are most probable,

(c) Either to extract a single mode with highest probability (based on the result of the classification).

In cases (a) and (b), the classification optimizes the list of coding modes candidates. The index of the optimal mode (based on the modified list) is transmitted in the bit stream. In the case (c), the most probable mode is selected; no index is thus needed to be signaled in the bit stream.

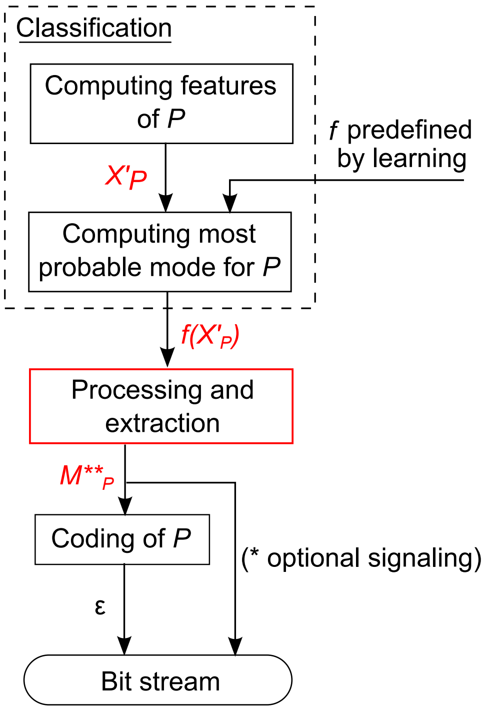


Figure 8.6 – *Proposed encoder scheme based on machine learning.*

The decoder corresponding to the proposed method, illustrated in figure 8.7, includes the following steps:

Step 1: Classifying P

- Calculation of causal characteristics X'_P of P
- Deduction of probable coding modes of P, by applying the classifier f, predefined a priori from decoded blocks as training samples, on X'_P. The result $f(X'_P)$ of this classification includes in general the list of several probable coding modes of P

Step 2: Processing and extraction of a coding mode M_P^{**} from $f(X'_P)$ that includes the list of probable coding modes of P and their probability.

Step 3: Decoding P with the computed mode M_P^{**} and the texture residual ϵ parsed from the bit stream. The index of M_P^{**} is optionally parsed depending on the processing and extraction module as mentioned below.

Given that the classification provides in general the list of several probable coding modes of P, the processing and extraction module at the decoder side allows:

(a) Either to modify the list of probable modes by reordering them in function of their probability, identically to that of the encoder,

(b) Either to modify the list of probable modes by removing modes with least probability, identically to that of the encoder,

(c) Either to extract the best mode with highest probability, identically to that of the encoder.

In the two cases (a) and (b), the classification optimizes the list of candidates modes. The index of the optimal mode (based on the modified list) is parsed from the bit stream. In the case (c), the most probable mode

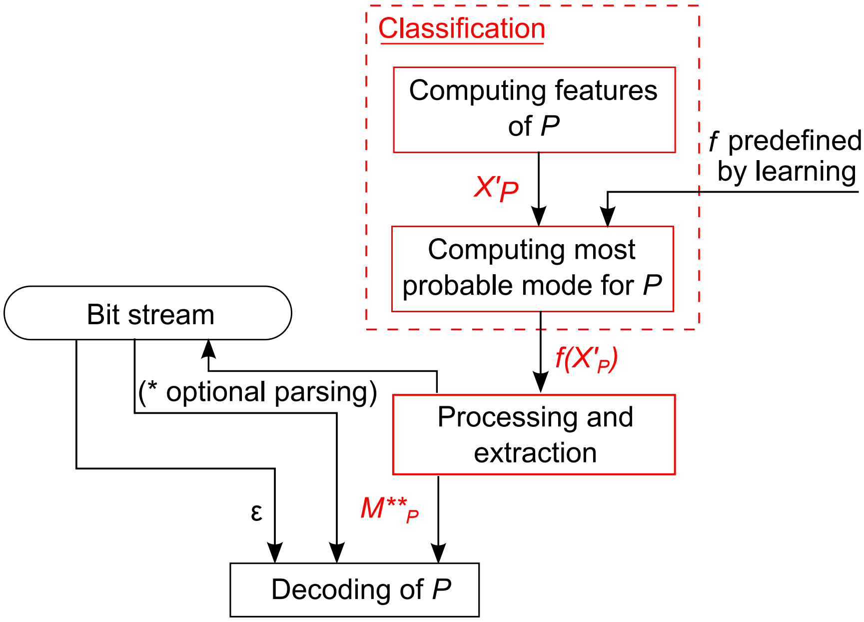


Figure 8.7 – *Proposed decoder scheme based on machine learning.*

is selected; thus no index needs to be read from the bit stream.

It is important to note that features of a block must be causal, i.e. constructed based on decoded data, so that similar learning process can be performed at decoder side.

In both diagrams shown in figures 8.6 and 8.7, the classifier f is predefined. It is a priori knowledge, independent of the current sequence to encode. We refer this type of classifier as static classifier.

8.2.2 Possible variants

We propose some variants for the general coding scheme:

Variant 1: The block classifier f, used in the training step to predict probable coding modes of blocks, is proposed to be computed on-the-fly in a process called *training step*. To differentiate with static block classifier f which is calculated from a predefined set of training samples, we denote this block classifier calculated on-the-fly by f'. This training step, depicted in figure 8.8, precedes the classification step in the general codec scheme and provides a classifier f' to be used in the blocks classification. There are two operations in this training stage:

- Calculation of causal features $X'_{P'_1}$, $X'_{P'_2}$,..., $X'_{P'_n}$ of causal blocks P'_1, P'_2,..., P'_n located in previously decoded frames
- Computation of classifier f' using machine learning techniques, based on the set $X'_{P'_1}$, $X'_{P'_2}$,..., $X'_{P'_n}$ of causal blocks P'_1, P'_2,..., P'_n and their optimal coding mode. If the optimal coding mode of a block P'_i is not available, a SDec competition of coding modes can be conducted on P'_i.

Figure 8.8 shows how f' is calculated to feed the entry of the classi-

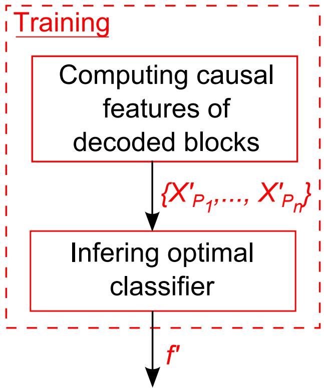


Figure 8.8 – *Additional training step proposed in variant 1.*

fication module. This on-the-fly computation of f' allows to adapt to the content of the video sequence. This type of classifier is referred to as dynamic classifier, which is not predetermined a priori but is a part of the encoding/decoding scheme.

At the decoder side corresponding to this proposed variant, a decoding scheme similar to the one illustrated in the figure 8.7 is used, except the addition of the training stage which precedes the classification stage.

Variant 2: Another variant is to use the classification in addition to the R-D competition of coding modes and to modify the mode transmission. In this variant, the difference between the optimal mode computed by the R-D competition and the predicted mode computed by the classification is signaled in the bit stream.

8.2.3 Features to be used in block classification

Concerning the block features to be exploited, for the sake of simplification, we propose following types of histograms to construct the feature descriptor of the current block. All the histograms are computed based on data of five causal neighboring blocks (Left, Above, AboveLeft, LeftBottom, AboveRight) as illustrated in the figure 8.9:

- Grayscale histogram: sampled to fit a 128-bins histogram.
- Histograms based on Gabor filters: consisting of four concatenated grayscale histograms, which are computed after applying four Gabor filters (Fogel and Sagi 1989) with angles respectively of 45°, 90°, 135°, 180° as illustrated in figure 8.10. Since each grayscale histogram corresponding to a Gabor angle has 128 bins, the concatenated histogram has 512 bins.
- Histogram of Oriented Gradients (HOG): a 36-bins histogram corresponding to 9 directions of gradient. HOG histograms are calculated using *VLFeat* toolbox (Vedaldi and Fulkerson 2010). The algorithm decomposes the image into square cells of a given size (typically 8×8 pixels), computes a histogram of oriented gradient in each cell, and then renormalizes the cells by looking into adjacent blocks.
- Optical flow (OF) based histograms: consisting of a 256-bins histogram of motion vectors (MV) magnitude coupled with a 360-bins

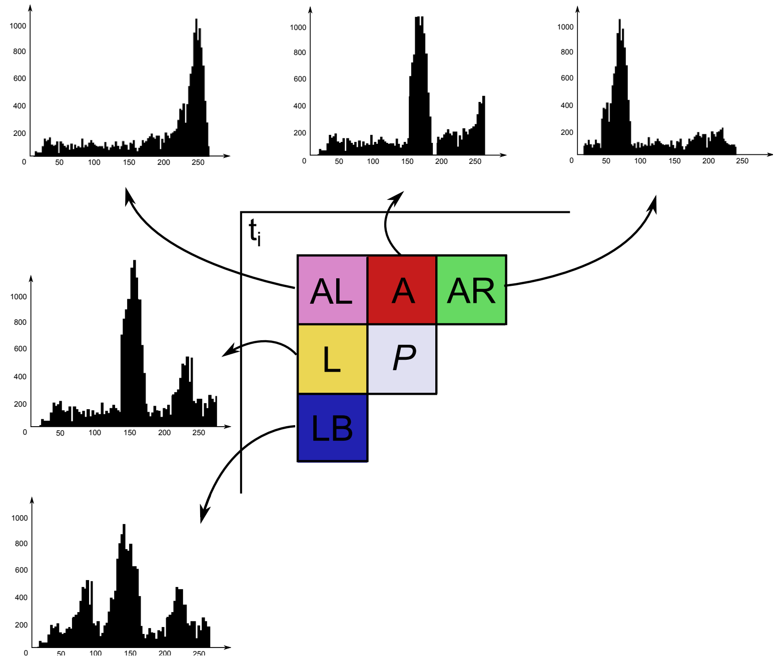


Figure 8.9 – *Computing histograms on causal neighboring blocks for the current block P in the frame at instant t_i.*

histogram of motion orientations. MVs are computed on pixel level using OF technique applied on causal regions in both previously decoded frame and current frame. The algorithm to calculate the OF can be found in (Liu 2009).

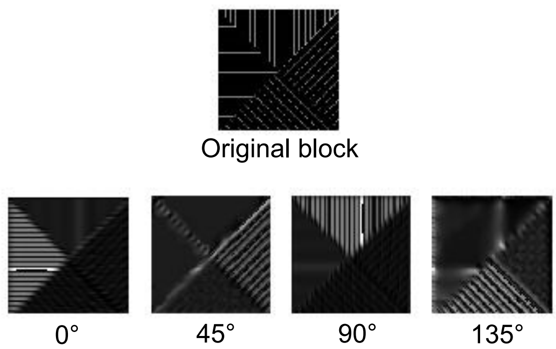


Figure 8.10 – *Different Gabor filters applied on a block.*

For each block, all those histograms are computed on all five neighboring blocks and are concatenated to produce a vector corresponding to the feature of the current block. In our practical applications, the size of the feature vector is 6460.

8.3 Proposed practical applications

To illustrate the proposed coding scheme, we present in this section two simple practical applications where classification algorithm is used to predict the optimal R-D coding choice for the current block. The R-D computation is also conducted along with the classification stage to compute the optimal coding mode, and the difference between the predicted mode and the optimal mode is signaled in the bit stream. It is in fact the variant 2 already described in the general description of the proposed method. We also tested the case where the choice predicted by the classification is directly used to encode current block without any R-D computation. The probability that the predicted choice is optimal is not high enough to provide the compression gain. This is due to the block classifier which still has a limited performance in predicting probable coding modes.

Additionaly, we limit the number of classes in the classification to three for simplification purpose. Furthermore, the processing and extraction module only considers a single coding mode which is the most probable mode predicted by the classification.

The specific encoding scheme of our proposed practical applications is illustrated in the figure 8.11:

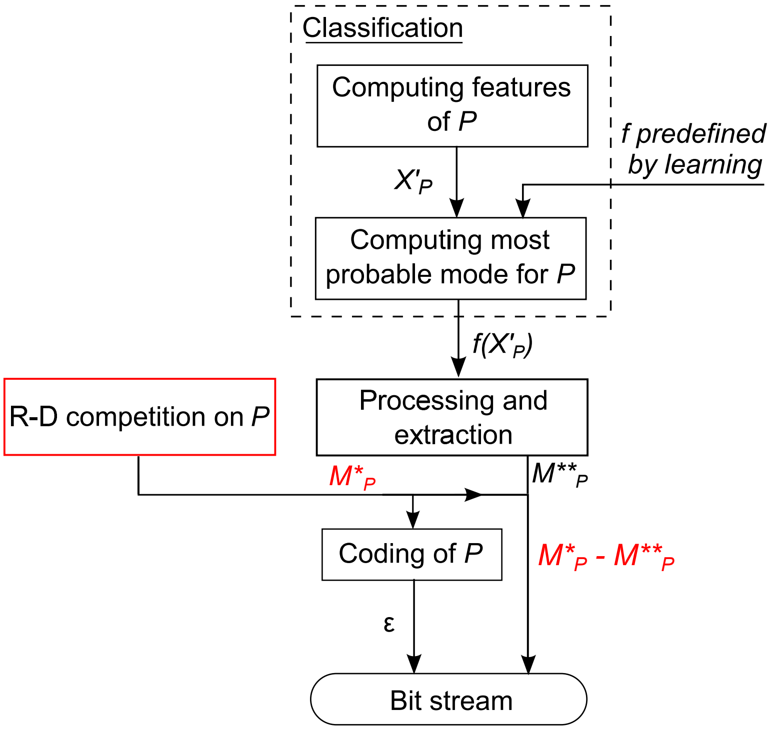


Figure 8.11 – *Encoding scheme of the proposed practical applications: the difference between the optimal mode computed by the R-D competition and the most probable mode predicted by the classification is signaled in the bit stream.*

We implemented the proposed scheme in the HEVC software test model version 12 (HM12). For the classification, we exploit Support Vector Machine (SVM), one of the most efficient machine learning algorithms. Being a powerful classifier, SVMs can be a useful tool when the data are not regularly distributed or have an unknown distribution. It is also often used to classify problems with arbitrary complexity. We choose SVM^{multiclass} (Joachims 1999) as the SVM implementation to be the core of our classification module. This SVM software is integrated directly in the test model to allow on-the-fly prediction.

In the general outline of the proposed scheme, there is no restriction on classification algorithms used for predicting block optimal R-D choice. Algorithms other than SVM can indeed be envisaged.

A preliminary study is made with the objective to determine the coding modes that are suitable to be predicted by the proposed approach. We compute in table 8.1 the signaling cost of different syntax elements mentioned in figure 8.12 which represents the HEVC coding modes signaling scheme. The results are obtained in configuration LD-P-Main.

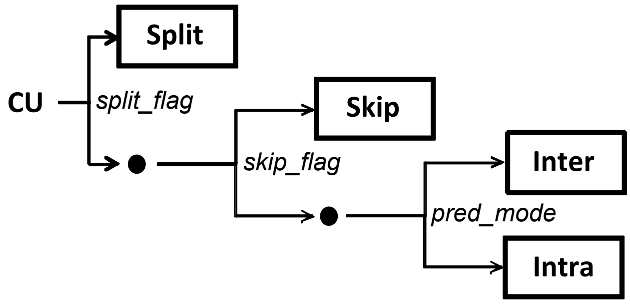


Figure 8.12 – *HEVC signaling scheme for different coding modes Split/Skip/Inter/Intra*

Sequences	***split_flag***	***skip_flag***	***pred_mode***
BasketBallPass	2.6	4.1	0.8
BQSquare	2.2	3.6	0.1
BlowingBubbles	2.0	2.6	1.2
RaceHorses	2.0	3.0	1.5
Average	**2.2**	**3.3**	**0.9**

Table 8.1 – *Cost of some syntax elements signaling coding modes in the total bit stream (%) in HEVC for some video sequences.*

We observe that if a block is not split, indicating whether the block is Skip or NoSkip by signaling $skip\_flag$ requires an average signaling overhead of 3.3% in the total bit stream. Signaling cost of $split\_flag$ and *pred_mode* is respectively 2.2% and 0.9%. Note that all those three syntax elements are encoded using CABAC entropy coding. The significant proportion of $skip\_flag$ in the total bit stream suggests us to predict Split/Skip/NoSkip in the proposed method since the potential impact on the coding performance is significant.

Another study, shown in table 8.2, evaluates the signaling cost of syntax elements related to the Intra MPM signaling scheme as illustrated in figure 8.13. Configuration AI-Main is used for this experiment.

We observe in table 8.2 that the syntax element $pred\_intra\_luma\_pred\_flag$, signaling whether or not the Intra direction to be transmitted is among three MPM values, costs 3.5% of the total bit stream on tested sequences. The syntax element mpm_idx, indicating if the first MPM value matches the Intra direction, costs 2.4% of total bit stream. Note that the syntax elements $pred\_intra\_luma\_pred\_flag$ is encoded using CABAC entropy coding, while mpm_idx is encoded with CABAC equiprobable mode (i.e. bypass mode). Since the proportion of these two syntax elements is significant, we propose in the second practical applica-

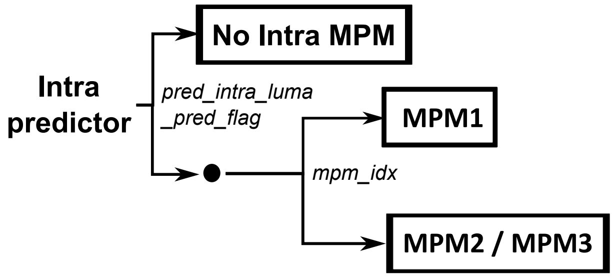


Figure 8.13 – *HEVC Intra MPM signaling scheme.*

tion where classification process is to predict following three classes: NoIntraMPM/MPM1/MPM2-MPM3.

Sequences	***pred_intra_luma_pred_flag***	***mpm_idx***
BasketBallPass	3.3	2.7
BQSquare	3.5	2.3
BlowingBubbles	3.9	2.3
RaceHorses	3.4	2.2
Average	**3.5**	**2.4**

Table 8.2 – *Cost of some syntax elements signaling Intra MPM in the total bit stream (%) in HEVC for some video sequences.*

8.3.1 Application for classifying coding modes

In this practical application, the machine learning based classification is exploited to predict the most probable mode between Split/Skip/NoSkip modes. Only LD-P-Main configuration is considered.

8.3.1.1 Signaling scheme

Two syntax elements are introduced as follows:

– *learn_opt_flag*: to signal whether or not the choice predicted by the classifier matches the optimal choice computed by the R-D competition.
– *learn_correction*: to correct the decision of the classifier in case *learn_opt_flag* is false by signaling the difference between the optimal choice and the choice predicted by the classifier.

With those syntax elements, the signaling scheme adapted for this proposed practical application is illustrated in figure 8.14, replacing the conventional scheme.

Note that, if the R-D optimal mode cannot be predicted correctly by the classifier, it is enough to signal it between two remaining modes using *learn_correction* with only one bit.

Compared with the HEVC signaling scheme, the gain comes from a Skip block that is correctly predicted (gain of one bit), and the loss comes from a Split block that is incorrectly predicted (loss of one bit).

For the decoding process, the same training and classification steps are performed for a block based on its causal feature. By reading *learn_opt_flag*

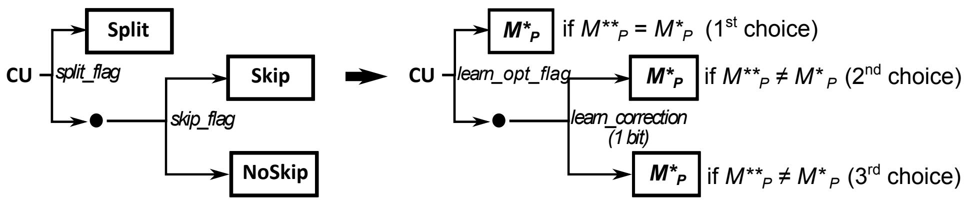


Figure 8.14 – *HEVC signaling scheme for Split/Skip/NoSkip (left) replaced by proposed machine learning based signaling scheme (right). Each of three choices takes a value among Split/Skip/NoSkip.*

and *learn_correction* (in case *learn_opt_flag* is false), the optimal R-D choice can be retrieved to decode the current block.

We design CABAC contexts for two newly introduced syntax elements *learn_opt_flag* and *learn_correction* as follows:

- *learn_opt_flag*: for each block size of 64×64, 32×32 or 16×16, 3 contexts are defined based on *learn_opt_flag* of the neighboring blocks, in a similar way as for *split_flag* or *skip_flag* of HM12 codec:
  - None of the Left and Above blocks is encoded with *learn_opt_flag* $= 1$ or is available
  - Only one block among the Left and Above blocks is encoded with *learn_opt_flag* $= 1$
  - Both the Left and Above blocks are encoded with *learn_opt_flag* $= 1$
- *learn_correction*: for each block size, 10 contexts are defined based on the combination of cases where coding modes of the neighboring Left and Above blocks take values among {Split/Skip/NoSkip/NotAvailable}.

For evaluation purposes, the proposed scheme is compared against the reference HEVC. Experimental results will be detailed in the next sections. Furthermore, we also give, as side information, the compression performance of proposed method on raw binary bits in comparison with HEVC. We need to disable CABAC contexts for related syntax elements, i.e. *learn_opt_flag*, *learn_correction* in the proposed scheme and *split_flag*, *skip_flag* in HEVC. Otherwise, CABAC contexts mechanism will further compress bits related to those syntax elements by exploiting local correlation with probability modelling.

8.3.1.2 Selection of features to be exploited

We perform a test to evaluate the efficiency of four types of histograms mentioned in previous section as block features. In table 8.3, each column and row corresponds to a type of histogram. Therefore, diagonal values represent the classifier accuracy when only one type of histogram is used. This accuracy value represents the performance of the block classifier, indicating how often the classifier makes correct prediction of the optimal coding mode for tested blocks. Other values correspond to the combination of two different types of histograms, where both are concatenated to produce a vector corresponding to the feature of the current block. Note that histograms are normalized with a L2-norm for the concatenation.

We observe that any case at position (i, j) will have better accuracy value than cases at positions (i, i) and (j, j) on the diagonal, proving that combining different histograms improves block classifier accuracy regarding the use of a unique histogram. Eventually, a combination of all four types yields best accuracy of 59.2%. .

	Grayscale	**Gabor**	**HOG**	**OF**
Grayscale	52.8%	56.8%	54.5%	56.2%
Gabor		53.7%	54.3%	53.7%
HOG			50.7%	53.3%
OF				50.7%
Combined use of all 4 types: 59.2%				

Table 8.3 – *Classifier accuracy when using different types of histograms as block features for classifying Split/Skip/NoSkip modes.*

8.3.1.3 Number of frames in training set

For the training set on which the block classifier is built, we propose to use the dynamic sliding window to adapt to the content of the tested sequences. Different number of reconstructed frames in the training set (i.e. window width) is tested: 1, 2, 4, 8 or 16 frames. The results are given in table 8.4, where CABAC contexts are disabled as previously mentioned in order to evaluate the raw bit rate saving. The combination of four proposed types of histograms is used as block feature.

Sequences	**1 fr.**	**2 fr.**	**4 fr.**	**8 fr.**	**16 fr.**
BasketBallPass	-0.5	-1.2	-2.1	-2.1	-1.6
BQSquare	-0.4	-0.5	-0.9	-1.2	-1.2
BlowingBubbles	-0.1	-0.2	-0.0	-0.1	0.5
RaceHorses	-0.3	-0.1	-0.2	-0.3	-0.4
Average	**-0.3**	**-0.5**	**-0.8**	**-0.9**	**-0.7**
Classifier accuracy	56%	58%	59%	59%	59%

Table 8.4 – *B-D rate savings (%) of proposed method for classifying Split/Skip/NoSkip modes with different widths of sliding training window (CABAC contexts disabled).*

We observe that increasing the number of reconstructed frames in the training set up to 8 frames improves the classifier accuracy which results in more coding gain. This confirms that the learning algorithm effectively exploits structural similarities in previously reconstructed frames to make predictions. However, taking too many frames in the training set (16 frames) reduces in turn the gain because the data from more distant frames is less correlated to the current frame. Note that most test sequences have frame rate of 30 Hz or 50 Hz.

8.3.1.4 Coding gain

All coding gains are measured using Bjøntegaard Delta (B-D) rate Bjøntegaard (2001) which represents the average difference between two R-D curves on the considered QP range. Standard QP values of 22, 27, 32 and 37 are used. Only the configuration LowDelay-P-Main is tested.

Coding performance of the proposed method is evaluated against the reference HEVC and is given in table 8.5. 8-frames sliding window is used. The comparison in raw bit rate reduction is also provided. There are thus two columns corresponding to cases where CABAC is respectively enabled and disabled for related syntax elements in both the tested version and the reference version. The theoretical maximum gain is also provided in brackets, which corresponds to the case where the classifier always predicts correctly the optimal coding mode of the current block (i.e. no correcting information is signaled).

	CABAC ctx. enabled	**CABAC ctx. disabled**
BasketBallPass_wvga	-0.3 (-2.6)	-2.1 (-2.5)
BQSquare_wvga	-1.0 (-4.1)	-1.2 (-2.8)
BlowingBubbles_wvga	-0.0 (-2.5)	-0.1 (-1.4)
RaceHorses_wvga	-0.2 (-3.4)	-0.3 (-1.3)
Anemone_wvga	-0.5 (-7.6)	-0.2 (-3.7)
Book_wvga	-0.1 (-6.6)	-2.5 (-1.2)
Ducks_wvga	-0.2 (-5.3)	-1.1 (-2.2)
Keiba3_wvga	-0.2 (-3.6)	-0.2 (-1.8)
Flower4_qwvga	-3.0 (-3.8)	-6.3 (-7.1)
Keiba3_qwvga	-0.5 (-2.9)	-0.4 (-1.4)
Nuts3_qwvga	-0.0 (-3.8)	-2.6 (-4.8)
HallMonitor_qwvga	-0.8 (-3.7)	-2.1 (-2.8)
Irene_qwvga	-0.4 (-3.1)	-1.7 (-2.9)
Marc_qwvga	-1.0 (-5.2)	-0.6 (-2.6)
Modo_qwvga	-3.6 (-9.2)	-6.0 (-6.1)
NewsCar_qwvga	-0.1 (-3.7)	-1.7 (-3.2)
Average	**-0.8 (-4.4)**	**-1.8 (-3.0)**

Table 8.5 – *B-D rate savings (%) of proposed method classifying Split/Skip/NoSkip modes with CABAC contexts enabled/disabled for related syntax elements. Theoretical maximum gain is given in brackets.*

When comparing with the reference HEVC, the performance of the proposed approach is interesting, providing an average gain of -0.8%. Up to -3.6% is achieved for the sequence "Modo_qwvga".

If we observe the comparison in terms of raw bit rate reduction, i.e. when CABAC contexts are disabled, the proposed approach can efficiently save up the signaling overhead dedicated for Split/Skip/NoSkip modes compared to HEVC. A significant coding gain of -1.8% in average is achieved in average. Gain up to -6.3% is reported. This result also proves that our CABAC contexts designed for *learn_opt_flag* and *learn_correction* in the proposed scheme do not perform as good as the contexts for *split_flag* and *skip_flag* in HEVC. Indeed, contexts for HEVC syntax elements are highly optimized to achieve such level of compression.

We also provide in table 8.6 the classifier accuracy specifically when predicting each mode among Split/Skip/NoSkip in case CABAC contexts are disabled. It is reported that the proposed classifier predicts less accurately NoSkip mode than Split and Skip modes. Effectively, NoSkip includes Inter and Intra modes which are often selected on blocks with less spatio-temporal correlation from previously reconstructed data than Skip mode. The prediction, which is based on blocks correlation, is thus more prone to error.

Predicted modes	% blocks	Prediction accuracy
Split	46.7%	77.6%
Skip	24.0%	63.8%
NoSkip	29.3%	26.0%
All	100%	59%

Table 8.6 – *Percentage of blocks and classifier accuracy for each predicted mode when classifying Split/Skip/NoSkip modes (CABAC contexts disabled).*

Finally, we note that the runtime in this practical application is very high and not yet suitable for realistic video coding due to following reasons:
- SVM^{multiclass} software: this SVM implementation is known to be very slow when used for classifying multiple classes.
- Computation of histograms as block features: the construction of different types of histograms (Gabor, HOG and OF) also requires significant processing time.

We also conducted experiments related to the classification of four or five classes by increasing the number of coding modes to be classified. Unfortunately, loss in compression performance compared to the classification of three classes is observed: there is no gain in average. This is mainly due to the decrease in performance of block classifier in correctly predicting block coding modes. Indeed, the correct prediction rate is lower compared to the case of classifying three classes. The classification accuracy appears to degrade as the number of classes increases.

8.3.2 Application for classifying Intra MPM flags

We present another practical application of the proposed coding scheme based on learning techniques, where flags related to the Intra MPM signaling scheme are predicted by the classification. Indeed, the Intra MPM scheme of a block is correlated to the neighboring blocks. Thus, it can be possible that, by learning how the Intra MPM scheme is exploited on a large set of blocks in function of the neighbors, given a block and its neighboring blocks, its Intra MPM scheme can be deduced.

Only AI-Main configuration is used for this experiment. Furthermore, for simplification purpose, we only apply the proposed scheme on $2N \times 2N$ blocks, which concern nearly 75% of the total number of all Intra blocks.

8.3.2.1 Signaling scheme

In this practical application, the signaling scheme adapted to the proposed method is illustrated in figure 8.15 and replaces the HEVC Intra MPM signaling scheme.

8.3.2.2 Selection of features to be exploited

In this practical application, we exploits all following histograms:
- Grayscale histogram
- Histograms based on Gabor filters

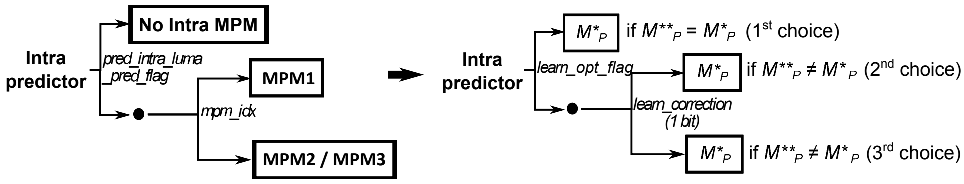


Figure 8.15 – *HEVC Intra MPM signaling scheme (left) replaced by proposed learning based signaling scheme (right). Each of three choices takes a value among NoIntraMPM, MPM1, or MPM2/MPM3.*

– Histogram of Oriented Gradients (HOG)

Tests are conducted in in AI-Main configuration, where all frames are encoded in Intra. Histograms based on optical flow are thus removed since this technique requires data of other frames.

8.3.2.3 Selection of training set

Considering a sequence to be encoded, in AI-Main configuration, since each Intra frame is encoded independently, we cannot exploit previously decoded frames to be used as the training set to infer the block classifier when encoding the current frame. We propose thus to use a predefined set of samples, which consists of all decoded blocks in the first frame of every sequences in the HEVC test set, as the training set. In consequence, the classifier that is used in this practical application is static. This classifier is computed offline and is provided to the encoder/decoder through an input file during the encoding/decoding process.

8.3.2.4 Coding gain

All coding gains are measured using Bjøntegaard Delta on standard QP values of 22, 27, 32 and 37. The experiment is conducted using AI-Main configuration on the second frame of all sequences of standard HEVC test set. As in the previous practical application, the coding performance is evaluated against the reference HEVC and the comparison in raw bit rate reduction is also given, resulting in two columns corresponding to whether CABAC contexts are enabled or disabled for the related syntax elements. The theoretical maximum gain is also provided in brackets.

According to table 8.7, despite the significant theoretical maximum gain, no gain in average is observed on the HEVC test set with class F excluded. The proposed practical application yields no gain on most tested sequences. There is an exception in class F for the sequence "*SlideShow*_1280×720" that provides a significant coding gain of -20.1%. This particular gain will be explained later in the analysis of the classifier accuracy.

The encoding and decoding times are also provided as side information given that they are very high, mainly due to the computation of different types of histograms during the block classification and to the multiclass classification process of SVM^{multiclass}, the selected implementation of SVM.

	CABAC ctx. enabled	CABAC ctx. disabled
Class A	0.0 (-2.3)	0.0 (-0.9)
Class B	0.0 (-2.9)	-0.1 (-1.2)
Class C	-0.1 (-2.2)	0.0 (-1.1)
Class D	0.2 (-1.6)	0.0 (-0.5)
Class E	0.0 (-4.2)	-0.2 (-1.9)
Average	**0.0 (-2.6)**	**-0.1 (-1.1)**
Class F	-5.1 (-6.6)	3.7 (-4.2)
Max.	-20.1 (-20.7)	-0.3 (-14.3)
EncTime	1854251%	
DecTime	7205339%	

Table 8.7 – *B-D rate savings (%) of proposed method when classifying NoIntraMPM/MPM1/MPM2-MPM3 with CABAC contexts enabled/disabled for related syntax elements. Theoretical maximum gain is given in brackets.*

Table 8.8 shows the classifier accuracy for each predicted modes among NoIntraMPM/MPM1/MPM2-MPM3. We observe that in general, on the standard HEVC test set, the classifier cannot distinguish correctly each different mode, thus predicting nearly all blocks as "NoIntraMPM". This results in a very low rate of correct prediction for "MPM1" and "MPM2/MPM3" modes. The accuracy of the classifier is only 36.5%. For comparison purposes, using the MPM signaling scheme, the optimal Intra direction of a block can be correctly predicted by the first MPM candidate with an accuracy of 52%.

Predicted modes	% blocks	Prediction accuracy
NoIntraMPM	35.9%	99.7%
MPM1	32.9%	1.3%
MPM2/MPM3	31.2%	0.2%
All	100%	36.5%

Table 8.8 – *Percentage of blocks and classifier accuracy for each predicted mode when classifying NoIntraMPM/MPM1/MPM2-MPM3 (CABAC contexts disabled).*

In the particular sequence $''SlideShow\_1280 \times 720''$ where its visual content favors the "NoIntraMPM" mode, by predicting nearly all blocks with "NoIntraMPM" mode, the classifier provides by chance more accurate predictions, resulting eventually in a compression gain.

The low prediction accuracy of the classifier originates from the low discriminating power of the exploited histograms, given that they are based on simple and conventional existing types of histogram which are not able to convey spatial structure information. First, histogram does not take into account the spatial information. Effectively, all spatial information is discarded and only feature occurrence counts are retained when constructing the histogram. For example, given two blocks that display a same object but in different positions, all the histograms that are based on grayscale, on Gabor filters or HOG provide the same result for both blocks. Secondly, there is the problem of uniqueness and robustness. For example, two different images with similar grayscale distribution give very similar histograms. Similarly, the images of the same view with different conditions of lighting create very different grayscale histograms. Proposed Gabor filters based histogram also has the same robustness problem since it is in fact the concatenation of different grayscale histograms corresponding to different Gabor angles.

8.4 Perspectives

In order to increase the coding performance of proposed method, it is crucial to improve the performance of the block classifier, which depends largely on the block features being considered. There are several proposals to improve the discriminating capability of histogram features. One consists to exploit multi-resolution histograms (Lazebnik et al. 2006), which incorporate spatial structure information by partitioning the image into increasingly fine sub-regions and computing histograms of local features found inside each sub-region. The resulting "spatial pyramid" is a simple and computationally efficient extension of an orderless bag-of-features image representation, and it shows significantly improved performance on challenging scene categorization tasks. Using this type of histogram allows therefore to better distinguish different classes, providng more correct predictions in classification.

An example of constructing a three-level pyramid is shown in figure 8.16: the image has three feature types, indicated by circles, diamonds, and crosses. At the top, we subdivide the image at three different levels of resolution. Next, for each level of resolution and each channel, we count the features that fall in each spatial bin and form histograms. Finally, we weight each spatial histogram.

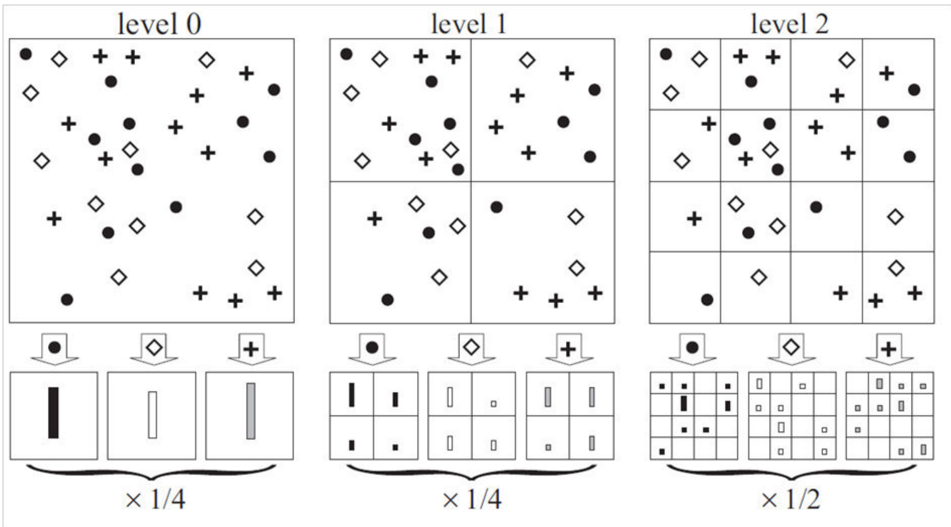


Figure 8.16 – *Example of constructing three-level pyramid multi-resolution histograms (Source: (Lazebnik et al. 2006)).*

Other methods propose to explicitly embed spatial structure information into histograms. Most notable examples are spatiograms (Birchfield and Rangarajan 2005) and correlograms (Huang et al. 1998).

Improving CABAC contexts for newly introduced syntax elements is also an interesting perspective, since the experimental results show that adapted contexts can significantly further improve the coding gain.

Finally, it is also important to reduce the runtime so that realistic video coding application can be made and to improve the performance of the learning and classification processes, for example by using other SVM implementations or classification algorithms other than SVM that are faster or more efficient.

Conclusion

In this chapter, a novel coding scheme that applies machine learning algorithms in video coding is presented. Unlike other existing works that mainly focus on reducing complexity, proposed approach allows improving compression ratio by reducing the signaling overhead. The general description of the approach is detailed along with different possible variants.

Practical applications of proposed method are also described, using SVM classifier to predict block coding modes (Split/Skip/NoSkip, NoIntraMPM/MPM1/MPM2-MPM3) based on block causal features descriptor. The descriptor consists solely of different types of histograms constructed based on causal neighboring blocks.

In the practical application that classifies Split/Skip/NoSkip coding modes, interesting bit rate saving is observed, with 0.8% of gain in average compared to the reference HEVC. In the other practical application that classifies NoIntraMPM/MPM1/MPM2-MPM3 related to the Intra MPM signaling scheme, no gain is achieved due to the low discriminating capability of the histograms being used for block classification.

Several perspectives are envisaged. The most interesting one consists to improve the classifier performance since the accuracy of classifier remains low in the proposed practical applications.

Conclusion and future works

Thesis objectives

This thesis studies the novel concept of Smart Decoder where the decoder is given the ability to simulate the encoder and is able to conduct the R-D competition similarly as in the encoder. The proposed technique aims to reduce the signaling of competing coding modes and parameters. In the first part of the thesis, the general outline of SDec encoding and decoding schemes is proposed, along with several practical applications where different parameters of the SDec scheme are studied. In the second part, a long-term approach, which is more research oriented, is introduced, presenting an innovative method that further makes use of the processing capacity of the SDec decoder. It is proposed that machine learning techniques, able to predict future outcome based on past data, are exploited in video coding with the purpose of reducing the signaling overhead.

Summary

SDec coding scheme

It is proved that reducing the signaling overhead of coding modes and parameters is crucial for next generations of video codecs. Therefore, the novel SDec coding scheme is proposed to reduce the signaling of competing coding modes and parameters by exploiting the processing ability of the decoder. Being calculated on the causal SDec reference block rather than directly on the current block, coding modes can be computed identically in both encoder and decoder sides. The transmission of those coding modes are thus not needed and can be removed. Using this SDec scheme, the integration of powerful tools that typically suffer from heavy signaling overhead is made possible.

Based on the general outline of the SDec coding scheme, a simplified practical application is proposed, using only Intra as the coding mode for the SDec competition. Up to 35 Intra directions are thus put in competition on the SDec reference block. An overall performance improvement is obtained despite different restrictions regarding the SDec general outline. On JCT-VC test set under CTC, the configuration using two candidates for the SDec reference yields an average bitrate savings of -0.4%, -0.5% and -0.9% respectively for AI, RA and LP configurations comparing to the standard HEVC reference software. The introduced complexity is however not negligible due to the fact that a competition of coding modes is conducted on the SDec reference block in both encoder and decoder sides.

Another practical application of the SDec coding scheme is also proposed, which exploits the Intra 1D coding mode. An implementation of Intra 1D is first given, then the SDec scheme is used to reduce the signaling of Intra 1D intrinsic parameters. Intra 1D mode is proved to be very efficient in coding sequences containing complex texture, with average gain of -3.2%, -0.8% and -0.1% respectively in AI, RA and LP configurations on a number of selected sequences. Indeed, compared to the classic Intra mode, the smaller block partitioning of Intra 1D allows to encode more accurately a block containing complex and non-linear details. When applying the SDec scheme, all intrinsic parameters of Intra 1D are competing on the SDec reference block and the optimal parameter is inherited to encode the current block. Further improvement is observed, compared with the use of Intra 1D without the SDec scheme, uniquely in LP configuration where temporal redundancy is exploited more efficiently by using colocated block as the SDec reference. Other methods selecting blocks to be the SDec reference can provide significant gain improvement but only for sequences with particular content. For example, template matching technique can be used to find the SDec reference block for sequences containing complex texture and translational motion.

Next, in another practical application of the SDec scheme, the SDec reference is re-encoded using a temporal coding mode during the SDec process. The motion parameters re-estimated on the SDec reference block are then inherited to encode the current block. The advantage of the method lays in the motion re-estimation which allows more accurate motion parameters. The motion re-estimation process can be based on block matching technique or optical flow technique. On a Merge block candidate considered as the SDec reference, the SDec scheme with block matching based motion re-estimation yields systematic gain, although not significant (0.2%), on HEVC test set in LP configuration.

The OF based motion re-estimation on the SDec reference, which provides the dense motion vector field to be inherited to encode the current block, is implemented in both 2D and 3D coding. Compared to the conventional motion vector, a dense motion vector field provides a better accuracy in motion prediction thanks to its finer granularity. In 2D, with temporal Merge candidate as the SDec reference, the scheme allows to achieve interesting gain on visioconferencing sequences, with -0.7% in average for class E of standard HEVC test set. In 3D, a more complex approach is proposed, introducing both the dense motion vector field computed by the OF technique and the motion vector computed by the SP-IVMP technique in a SDec competition on the SDec reference block located in the base view. Significant bit rate reductions of 2.4% and 2.5% for two dependent views, and 0.7% for coded and synthesized views are achieved.

Finally, we implement a practical application of the SDec scheme that exploits a multitude of coding modes during the SDec process. All the

Intra mode, the Intra 1D mode and the re-estimated motion are put in the competition on the SDec reference block. Experiments are conducted in both 2D and 3D video coding. In 2D, although no improvement in average gain is observed on the standard HEVC test set, the combined use of different coding modes in the SDec competition improves the efficiency of the SDec scheme on a wider set of sequences, allowing to better encode more sequences with particular characteristic and texture. In 3D, using an efficient block in the base view as the SDec reference, improved gains are observed. The complexity is however still high due to the nature of the SDec scheme. Moreover, the choice of block to be the SDec reference is still limited. Improvement using more sophisticated techniques, such as learning machine, is proposed to adaptively select the SDec reference and the coding modes being used in the SDec process depending on the visual content of the encoded sequences.

There are also works that aim to improve the Intra MPM signaling scheme, suggesting the addition of a fourth MPM candidate among a predefined set of proposed candidates. The implementation of this approach gives slight but systematic gains (0.1%), especially for All-Intra configuration, with negligible increase in runtime. The perspective to adaptively select a most probable candidate instead of predefined candidate motivates us to exploit machine learning techniques to provide predictions based on past data collected on already reconstructed frames.

Machine learning in video coding

For the long-term approach, our research focuses on exploiting machine learning in video coding to reduce the signaling overhead, unlike other existing works that mainly focus on reducing the complexity. A novel coding scheme is presented where predictions are made using machine learning techniques in both encoder and decoder sides, allowing to skip signaling the coding information. Practical applications of the proposed method are given using SVM classifier to predict block coding modes based on the block features descriptor, which consists of different types of histograms constructed from data of causal neighboring blocks.

In a practical application that predicts the most probable coding mode among Split/Skip/NoSkip, experimental results in LP configuration show that proposed coding scheme achieves good bit rate savings, with 0.8% of average gain on a set of tested sequences compared to the reference HEVC.

Although the obtained experimental result is interesting since it proves the benefit of exploiting machine learning in video coding, there are still several points to be improved, especially the low discriminating capability of the histograms being used for block classification since we only used simple histograms that do not convey spatial structure information.

Perspectives for future work

To further continue the research done in this thesis, several interesting perspectives are proposed. The first primary point concerns the improvement of the selection of the block to be the SDec reference by exploiting techniques that are more sophisticated. It is proposed for example to use machine learning technique, a concept that is introduced later as long-term approach, to adaptively select the SDec reference according to the coding modes being used during the SDec process of competition. Further works can be conducted to implement this idea.

For the long-term approach that consists of exploiting the prediction ability of machine learning applied in video coding, the remaining crucial point is to improve the performance of block classifier since the accuracy of the classifier remains low in the proposed practical applications, resulting in a low rate of correct prediction. Given that the performance of a classifier depends largely on the block features being considered, it is proposed to exploit more efficient histogram features which have better discriminating capability. Several proposals are described. One method consists to exploit multi-resolution histograms. Other method proposes to explicitly embed spatial structure information into histograms, such as with spatiograms and correlograms. Classification algorithms other than SVM can also be exploited.

Publications

Conference papers

1. D.-K. Vo-Nguyen, J. Jung, J.-M Thiesse and M. Antonini, "Schéma de compression vidéo basé sur l,exploitation d,un décodeur intelligent", Proceeding of COmpression et REprésentation des Signaux Audiovisuels (CORESA), Le Creusot, France, 2013
2. D.-K. Vo-Nguyen, J. Jung, J.-M Thiesse and M. Antonini, "Smart decoder: A new paradigm for video coding", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 2014

Patent

J. Jung, D.-K. Vo-Nguyen and M. Antonini, "Codage d'image et de vidéo par apprentissage et classification", 2015 (Patent pending)

Bibliography

J. An, Y. Chen, J.-L. Lin, Y.-W. Huang, and S. Lei. Inter-view motion prediction for hevc-based 3d video coding. *ITU-T SG 16 WP 3 & ISO/IEC JTC 1/SC 29/WG 11 JCT3V-A0049*, 2012. (Cited in 17.)

J. An, K. Zhang, J.-L. Lin, and S. Lei. Sub-pu level inter-view motion prediction. *ITU-T SG 16 WP 3 & ISO/IEC JTC 1/SC 29/WG 11 JCT3V-F0110*, October 2013. (Cited in 17.)

B. Bai, L. Cheng, C. Lei, P. Boulanger, and J. Harms. Learning-based multiview video coding. *Picture Coding Symposium (PCS)*, 2009. (Cited in xii and 25.)

S. T. Birchfield and S. Rangarajan. Spatiograms versus histograms for region-based tracking. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005*, pages 1158–1163, June 2005. (Cited in 156.)

G. Bjøntegaard. Calculation of average PSNR differences between RD curves. *VCEG-M33, ITU-T VCEG, 13th Meeting, Austin, TX, USA*, April 2001. (Cited in 48 and 151.)

X. Cao, C. Lai, Y. Wang, L. Liu, J. Zheng, and Y. He. Short distance intra coding scheme for high efficiency video coding. *IEEE Transactions on Image Processing*, 22(2):790–801, 2013. ISSN 1057-7149. (Cited in 38.)

P. Carrillo, T. Pin, and H. Kalva. Low complexity h.264 video encoder design using machine learning techniques. *International Conference on Consumer Electronics(ICCE)*, 2010. (Cited in 25 and 137.)

L. Cheng and S. V. N. Vishwanathan. Learning to compress images and videos. *Proceedings of the 24th International Conference on Machine Learning*, 2007. (Cited in 25.)

C-.K. Chiang, W.-H. Pan, C. Hwang, S.-S. Zhuang, and S.-H. Lai. Fast h.264 encoding based on statistical learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 2011. (Cited in 26 and 137.)

Cisco. Cisco visual networking index: Forecast and methodology, 2014-2019. *http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white-paper-c11-481360.html*, 2015. (Cited in 1.)

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 1995. (Cited in 24.)

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005*, pages 886–893, June 2005. (Cited in 140.)

C. Di and C. Yuan. A novel fast mode decision algorithm for h.264/avc based on probabilistic learning. *3rd International Congress on Image and Signal Processing*, 2010. (Cited in 25 and 137.)

T. T. Do, X. Lu, and J. Sole. Compressive sensing with adaptive pixel domain reconstruction for block-based video coding. *IEEE International Conference on Image Processing (ICIP)*, pages 3377–3380, 2010. ISSN 1522-4880. (Cited in 38.)

F. Dufaux and F. Moscheni. Motion estimation techniques for digital tv: A review and new contribution. *Proc. IEEE*, pages 858–876, 1995. (Cited in 22.)

O. D. Escoda, P. Yin, C. Dai, and X. Li. Geometry-adaptive block partitioning for video coding. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1:657–660, 2007. ISSN 1520-6149. (Cited in 38.)

I. Fogel and D. Sagi. Gabor filters as texture discriminator. *Biological Cybernetics*, 1989. (Cited in 145.)

D. Han, T. Purushotham, K.V.S. Swaroop, and K.R. Rao. Low complexity h.264 encoder using machine learning. *Signal Processing Algorithms, Architectures, Arrangements, and Applications Conference Proceedings (SPA)*, 2010. (Cited in 25 and 137.)

P. Helle, S. Oudin, B. Bross, D. Marpe, M.O. Bici, K. Ugur, J. Jung, G. Clare, and T. Wiegand. Block merging for quadtree-based partitioning in hevc. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12), December 2012. (Cited in xi, xiii, 12, and 95.)

B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981. (Cited in 22.)

J. Huang, S. R. Kumar, M. Mitra, and W.-J. Zhu. Spatial color indexing and applications. *International Conference on Computer Vision, 1998*, pages 602–607, January 1998. (Cited in 156.)

ITU-T. H.264 : Advanced video coding for generic audiovisual services. *ITU-T Recommendation H.264*, May 2003. http://www.itu.int/rec/T-REC-H.265-201304-I. (Cited in 7.)

ITU-T. High efficiency video coding. *ITU-T Recommendation H.265*, April 2013. http://www.itu.int/rec/T-REC-H.265-201304-I. (Cited in 33.)

ITU-T and ISO/IEC-JTC1. Generic coding of moving pictures and associated audio information - part 2: Video. *ITU-T Recommendation H.262 and ISO/IEC 13818-2 (MPEG-2 Video)*, 1994. (Cited in 7.)

R. Jillani and H. Kalva. Low complexity intra mb encoding in avc/h.264. *IEEE Trans. Consumer Electronics*, pages 277–285, 2009. (Cited in 25 and 137.)

T. Joachims. Making large-scale svm learning practical. *Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.)*, 1999. (Cited in 147.)

J. Jung and G. Laroche. Competition-based scheme for motion vector selection and coding. *VCEG-AC06*, July 2006. (Cited in 33.)

H. Kalva and L. Christodoulou. Using machine learning for fast intra mb coding in h.264. *Visual Communications and Image Processing*, 2007. (Cited in 25 and 137.)

S. Kamp, J. Ballé, and M. Wien. Multihypothesis prediction using decoder side motion vector derivation in inter frame video coding. *Proc. SPIE Visual Communications and Image Processing (VCIP)*, 9, 2009. (Cited in 20.)

S. Kamp, M. Evertz, and M. Wien. Decoder side motion vector derivation for inter frame video coding. *IEEE International Conference on Image Processing (ICIP)*, pages 1120–1123, 2008. ISSN 1522-4880. (Cited in 20.)

R. Krishnamurthy, P. Moulin, and J. Woods. Optical flow techniques applied to video coding. *Proc. International Conference on Image Processing*, pages 570–573, 1995. (Cited in 22.)

C. H. Lampert. Machine learning for video compression: Macroblock mode decision. *Proceedings of the 18th International Conference on Pattern Recognition*, 2006. (Cited in 25 and 137.)

G. Laroche. *Modules de Codage par Competition et Suppression de l'Information de Competition pour le Codage de Séquences Video*. PhD thesis, Telecom ParisTech, May 2009. (Cited in 33.)

G. Laroche, J. Jung, and B. Pesquet. Intra prediction with 1d macroblock partitioning for image and video coding. *Proc. SPIE 7257, Visual Communications and Image Processing(VCIP)*, 2009. (Cited in xii, 21, and 22.)

S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006*, pages 2169–2178, 2006. (Cited in xv and 156.)

X. Li, L. Zhang, and Y. Chen. Advanced residual prediction in 3d-hevc. *IEEE International Conference on Image Processing (ICIP)*, pages 1747–1751, 2013. (Cited in 18.)

C. Liu. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Massachusetts Institute of Technology, May 2009. (Cited in 101 and 146.)

C. Liu. Optical flow matlab/c++ code. *http://people.csail.mit.edu/celiu/OpticalFlow/*, 2012. (Cited in 109.)

D. G. Lowe. Object recognition from local scale-invariant features. *International Conference on Computer Vision*, pages 1150–1157, 1999. (Cited in 140.)

B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *Proceedings of the DARPA Image Understanding Workshop*, pages 121–130, 1981. (Cited in 22.)

W. Ma, S. Yang, L. Gao, C. Pei, and S. Yan. Fast mode selection scheme for h.264/avc inter prediction based on statistical learning method. *IEEE International Conference on Multimedia and Expo*, 2009. (Cited in 25 and 137.)

D. Marpe, H. Schwarz, and T. Wiegand. Context-based adaptive binary arithmetic coding in the h.264/avc video compression standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7), July 2003. (Cited in 13.)

E. G. Mora. *Codage multi-vues multi-profondeur pour de nouveaux services multimédia*. PhD thesis, TELECOM ParisTech, February 2014. (Cited in 23 and 105.)

K. Muller et al. 3d high-efficiency video coding for multi-view video and depth data. *IEEE Transactions on Image Processing*, 2013. (Cited in 16.)

K. Muller, P. Merkle, G. Tech, and T. Wiegand. 3d video coding with depth modeling modes and view synthesis optimization. *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1–4, December 2012. (Cited in xii and 42.)

T. Ojala, M. Pietikäinen, and D. Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. *International Conference on Pattern Recognition, 1994*, pages 582–585, 1994. (Cited in 140.)

T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *International Conference on Pattern Recognition, 1996*, pages 51–59, 1996. (Cited in 140.)

D. Rusanovsky, K. Muller, and A. Vetro. (Cited in 106 and 119.)

Y. Q. Shi, S. Lin, and Y. Q. Zhang. An optical flow based motion compensation algorithms for very low bit rate video coding. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 4:2869–2872, 1997. (Cited in 22.)

K. Sugimoto, M. Kobayashi, Y. Suzuki, S. Kato, and C.S. Boon. Inter frame coding with template matching spatio-temporal prediction. *IEEE International Conference on Image Processing (ICIP)*, 1:465–468, 2004. ISSN 1522-4880. (Cited in 20.)

G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12), December 2012. (Cited in xi, 7, 8, 10, and 12.)

T. K. Tan, C. S. Boon, and Y. Suzuki. Intra prediction by template matching. *IEEE International Conference on Image Processing (ICIP)*, pages 1693–1696, 2006. ISSN 1522-4880. (Cited in 20.)

G. Tech. Htm9.3 software. *https://hevc.hhi.fraunhofer.de/svn*, 2012. (Cited in 105.)

J.-M. Thiesse. *Codage Vidéo Flexible par Association d'un Décodeur Intelligent et d'un Encodeur Basé Optimisation Débit-Distorsion*. PhD thesis, Nice Sophia-Antipolis University, April 2012. (Cited in 20, 33, 34, and 38.)

J.-M. Thiesse, J. Jung, and M. Antonini. Hybrid-1D macroblock prediction for video compression. *17th European Signal Processing Conference*, pages 554–558, 2009. (Cited in xi, 22, and 38.)

H. R. Tohidypour, M. T. Pourazad, and P. Nasiopoulos. A low complexity mode decision approach for hevc-based 3d video coding using a bayesian method. *IEEE International Conference on Acoustic, Speech and Signal Processing*, 2014. (Cited in 25 and 137.)

B. Tomasik, P. Thiha, and D. Turnbull. Tagging products using image classification. *Proceedings of Special Interest Group on Information Retrieval (SIGIR), 2009*, July 2009. (Cited in xiv and 141.)

A. Vedaldi and B. Fulkerson. Vlfeat: an open and portable library of computer vision algorithms. *Proceedings of the international conference on Multimedia*, pages 1469–1472, October 2010. (Cited in 145.)

J. Xiong and H. Li. Fast and efficient prediction unit size selection for hevc intra prediction. *IEEE International Symposium on Intelligent Signal Processing and Communication Systems*, 2012. (Cited in 26 and 137.)

L. Zhang, Y. Chen, V. Thirumalai, J.-L. Lin, Y.-W. Chen, J. An, S. Lei, L. Guillo, T. Guionnet, and C. Guillemot. Inter-view motion prediction in 3d-hevc. *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 17–20, 2014. (Cited in 17.)

X. Zhou, C. Yuan, C. Li, and Y. Zhong. Fast mode decision for p-slices in h.264/avc based on probabilistic learning. *11th International Conference on Advanced Communication Technology*, 2009. (Cited in 25 and 137.)

Compression vidéo basée sur l'exploitation d'un décodeur intelligent

Dang Khoa VO NGUYEN

RESUME:

Cette thèse de doctorat étudie le nouveau concept de décodeur intelligent (SDec) dans lequel le décodeur est doté de la possibilité de simuler l'encodeur et est capable de mener la compétition R-D de la même manière qu'au niveau de l'encodeur. Cette technique vise à réduire la signalisation des modes et des paramètres de codage en compétition. Le schéma général de codage SDec ainsi que plusieurs applications pratiques sont proposés, suivis d'une approche en amont qui exploite l'apprentissage automatique pour le codage vidéo.

Le schéma de codage SDec exploite un décodeur complexe capable de reproduire le choix de l'encodeur calculé sur des blocs de référence causaux, éliminant ainsi la nécessité de signaler les modes de codage et les paramètres associés. Plusieurs applications pratiques du schéma SDec sont testées, en utilisant différents modes de codage lors de la compétition sur les blocs de référence. Malgré un choix encore simple et limité des blocs de référence, les gains intéressants sont observés.

La recherche en amont présente une méthode innovante qui permet d'exploiter davantage la capacité de traitement d'un décodeur. Les techniques d'apprentissage automatique sont exploitées pour but de réduire la signalisation. Les applications pratiques sont données, utilisant un classificateur basé sur les machines à vecteurs de support pour prédire les modes de codage d'un bloc. La classification des blocs utilise des descripteurs causaux qui sont formés à partir de différents types d'histogrammes. Des gains significatifs en débit sont obtenus, confirmant ainsi le potentiel de l'approche.

MOTS-CLES: Compression vidéo, HEVC, Décodeur intelligent, Codage intra, Codage Inter, Flux optique, Apprentissage automatique.

ABSTRACT:

This Ph.D. thesis studies the novel concept of Smart Decoder (SDec) where the decoder is given the ability to simulate the encoder and is able to conduct the R-D competition similarly as in the encoder. The proposed technique aims to reduce the signaling of competing coding modes and parameters. The general SDec coding scheme and several practical applications are proposed, followed by a long-term approach exploiting machine learning concept in video coding.

The SDec coding scheme exploits a complex decoder able to reproduce the choice of the encoder based on causal references, eliminating thus the need to signal coding modes and associated parameters. Several practical applications of the general outline of the SDec scheme are tested, using different coding modes during the competition on the reference blocs. Despite the choice for the SDec reference block being still simple and limited, interesting gains are observed.

The long-term research presents an innovative method that further makes use of the processing capacity of the decoder. Machine learning techniques are exploited in video coding with the purpose of reducing the signaling overhead. Practical applications are given, using a classifier based on support vector machine to predict coding modes of a block. The block classification uses causal descriptors which consist of different types of histograms. Significant bit rate savings are obtained, which confirms the potential of the approach.

KEYWORDS: Video compression, HEVC, Smart decoder, Intra coding, Inter coding, Optical flow, Machine learning.